NTNU – Trondheim
Norwegian University of
Science and Technology

Evolvability, Complexity and Scalability of Cellular Evolutionary and Developmental Systems

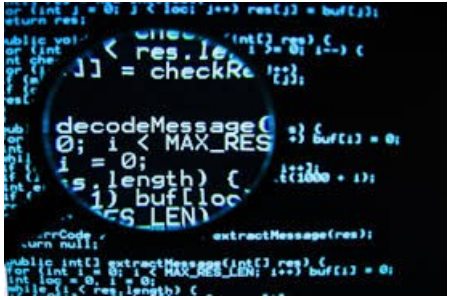
Stefano Nichele

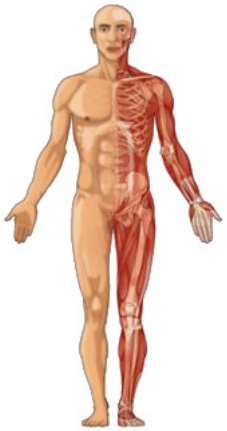
Outline

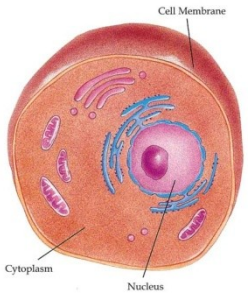
- Motivation / Introduction
- Research Questions
- Background
- Results (some)
  - Evolvability
  - Complexity
  - Scalability
- Conclusions and Further Work

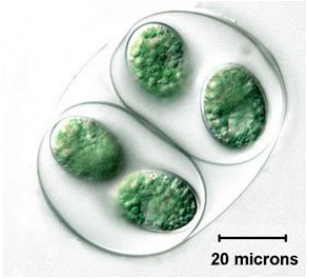
Motivation

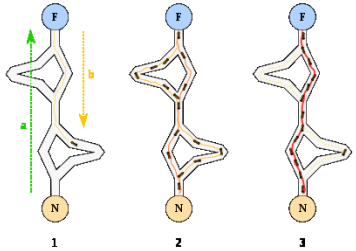
Motivation



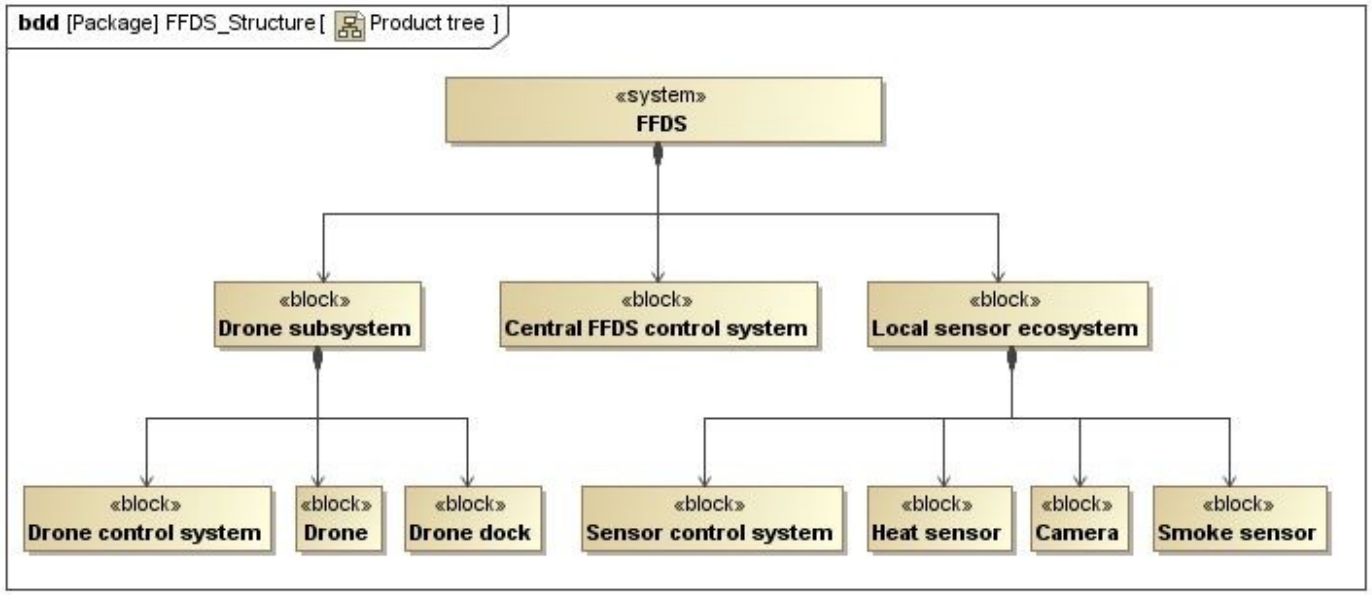




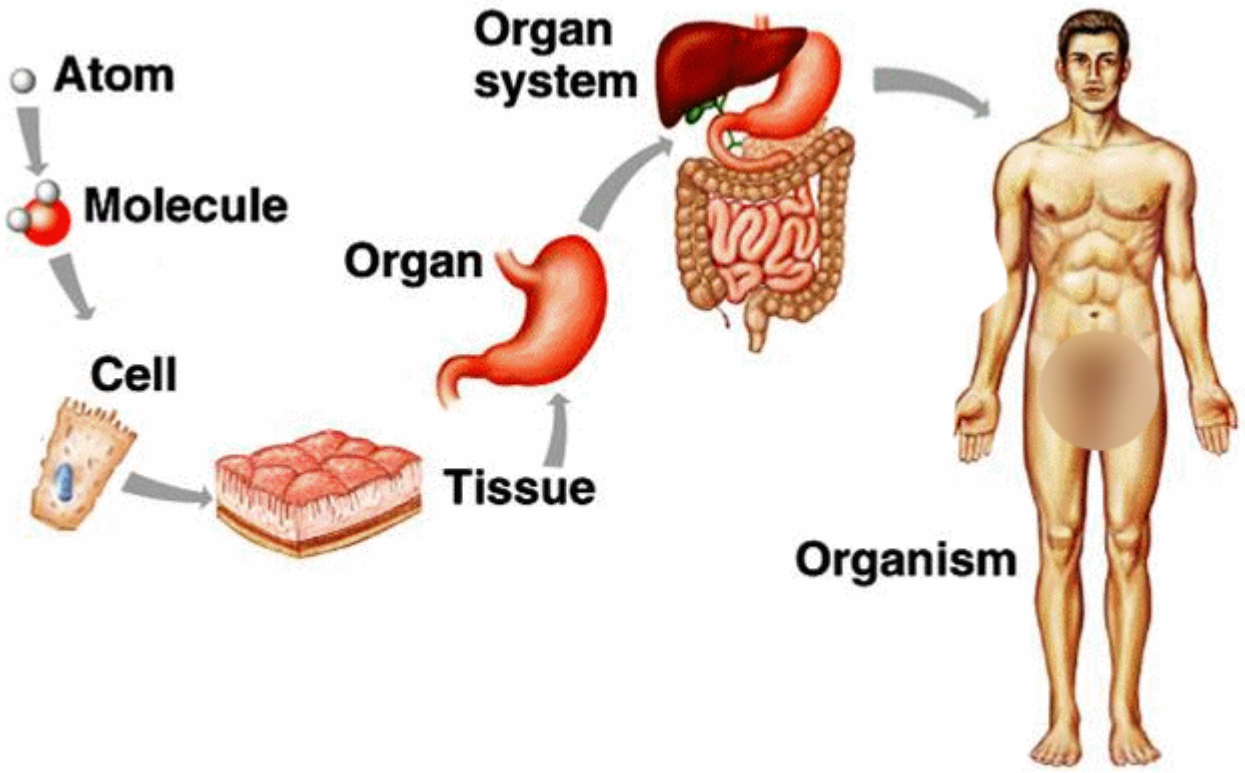




Engineering: top-down

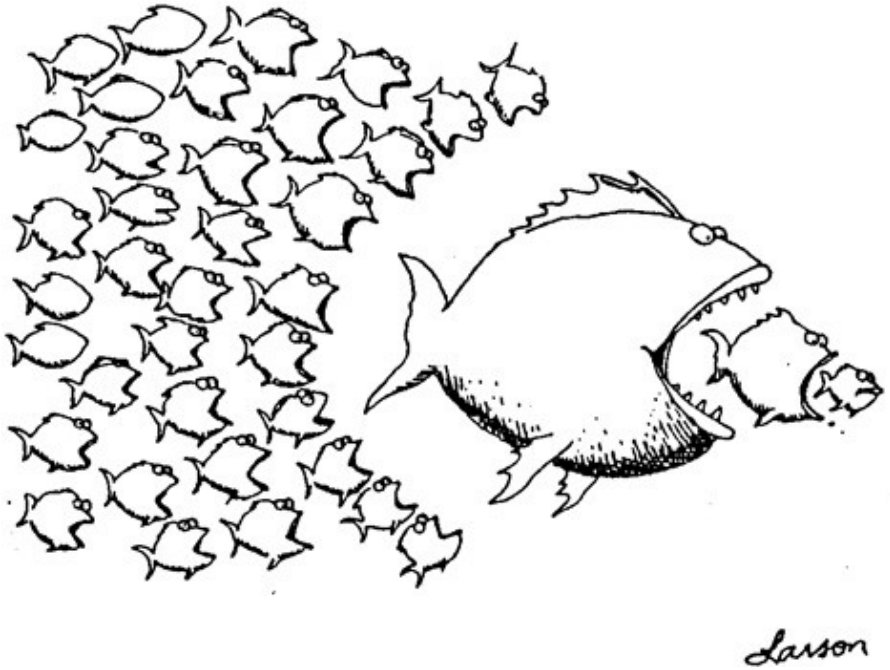


Nature: bottom-up



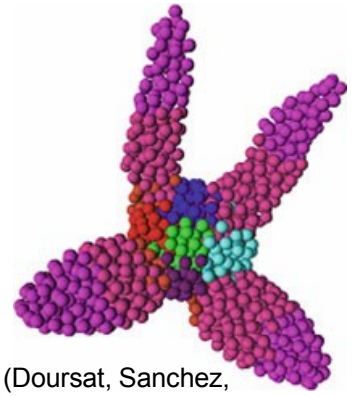
Conventional Engineering: attempts to *analyze* (top-down) systems

Bio-Inspired Computation: attempts to *synthesize* (bottom-up) life-like behaviors within computers and other artificial media

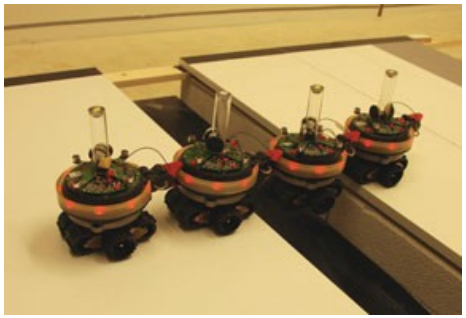


Emergent Complexity

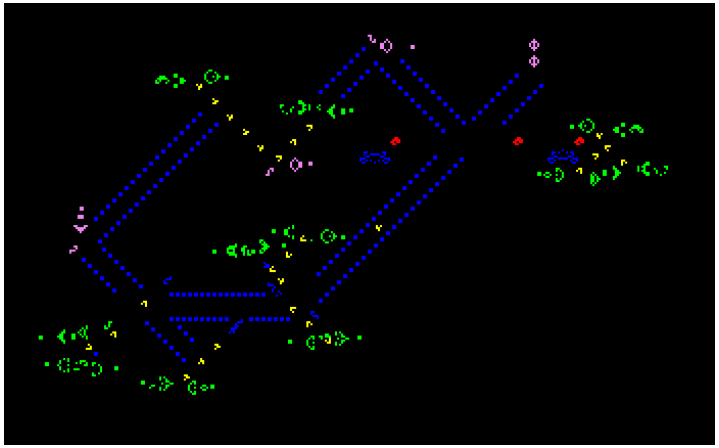
Examples: bio-inspired



(Doursat, Sanchez, Dordea, Fourquet, Kowaliw 2014)

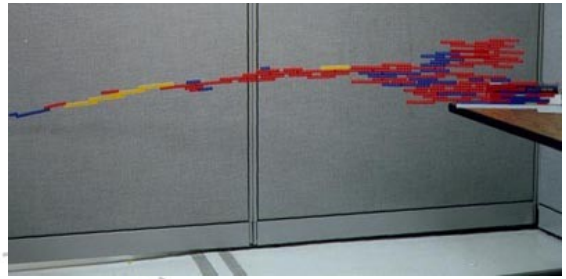


(Christensen, Grady, Dorigo 2009)



(Conway 1970)

(Hornby, Al Globus, Linden, Lohn 2006)

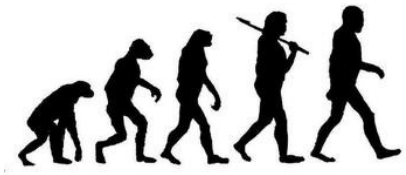


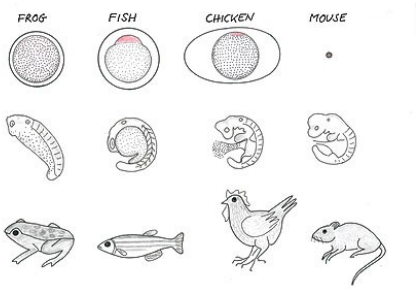
(Funes 1997)

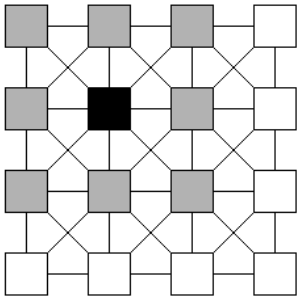
NTNU – Trondheim
Norwegian University of Science and Technology

Research Focus

- How to apply artificial evolution and development for the design of cellular machines that can produce complex computation and modelling?







Research Questions

Research Questions

RQ1:

- What kind of information must be present in the genome in order to produce computation in any of the computational classes?

Universality classes: CA computational behavior (Wolfram)

- What information must be present in the genome?
- What information processing capability must be available in the gene regulation?
- What cellular actions are required to be expressed as to be able to develop a target organism?

Research Questions

RQ2:

- How to quantify developmental complexity, i.e. emergent phenotypic complexity?

Development process as a whole

Phenotypic changes: trajectory, transient, attractor

Research Questions

RQ3:

- Do genome parameters give any information on the evolvability of the system? And if yes, can genome information be used to guide evolutionary search in favourable areas of the search space where the wanted emergent behavior is more likely to be found?

Research Questions

RQ4:

- How can scalability of artificial EvoDevo systems be improved towards achieving systems that can fully unleish their inherent complexity, e.g. potentially at the levels of complexity found in nature?

  Gene duplication

  Open ended

Contributions

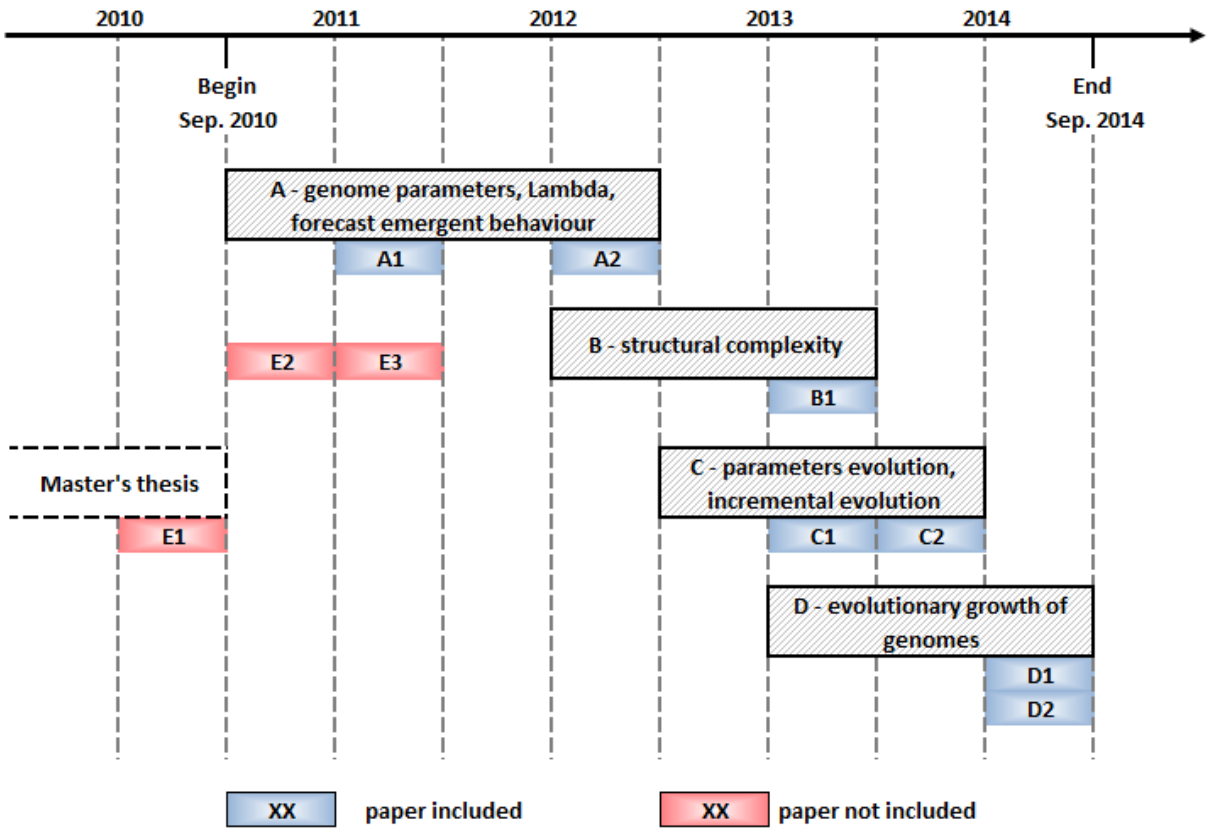
Paper N.	Title	Category
1	On the Correlations Between Developmental Diversity and Genomic Composition	A.1
2	Genome Parameters as Information to Forecast Emergent Developmental Behaviors	A.2
3	Measuring Phenotypic Structural Complexity of Artificial Cellular Organisms	B.1
4	Evolution of Incremental Complex Behavior on Cellular Machines	C.1
5	Investigation of Genome Parameters and Sub-Transitions to Guide Evolution of Artificial Cellular Organisms	C.2
6	Evolutionary Growth of Genome Representations on Artificial Cellular Organisms with Indirect Encodings	D.1
7	Evolutionary Growth of Genomes for the Development and Replication of Multicellular Organisms with Indirect Encodings	D.2
8	Trajectories and Attractors as Specification for the Evolution of Behavior in Cellular Automata	E.1
9	Discrete Dynamics of Cellular Machines: Specification and Interpretation	E.2
10	On the Edge of Chaos and Possible Correlations Between Behavior and Cellular Regulative Properties	E.3

Complexity

Evolvability

Scalability

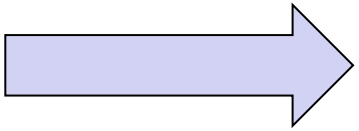
Chronological Structure



Background

Background

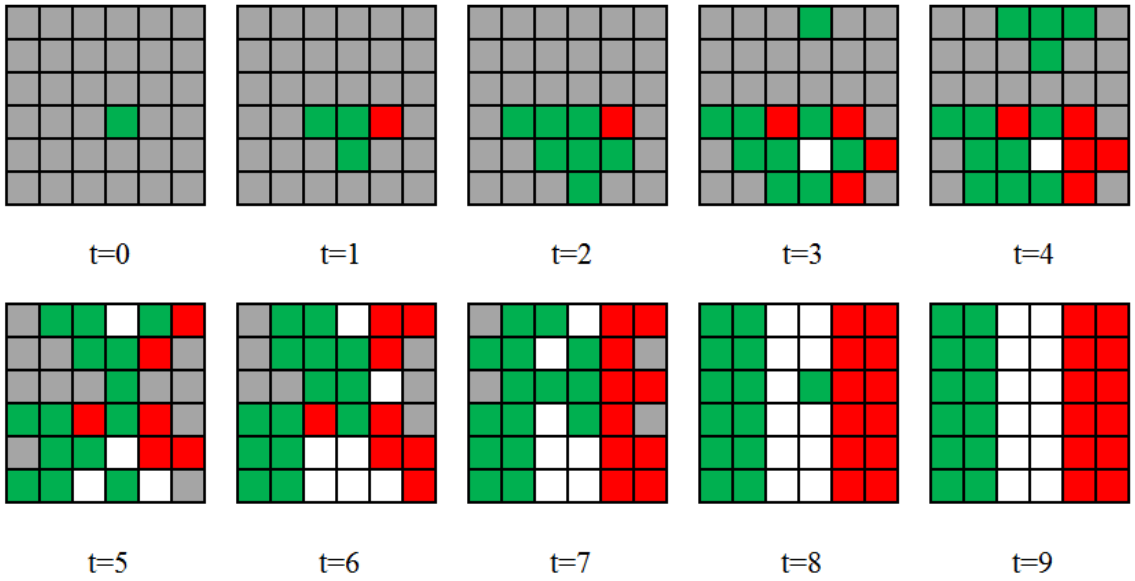
- Development
- Evolution
- Cellular Automata
- Genotype-Phenotype mapping & representations
- Genome parameters
- Complexification

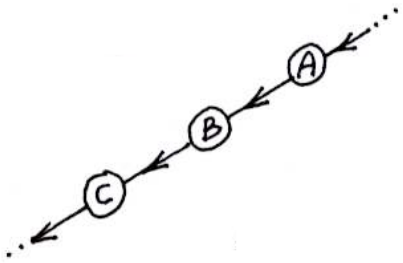


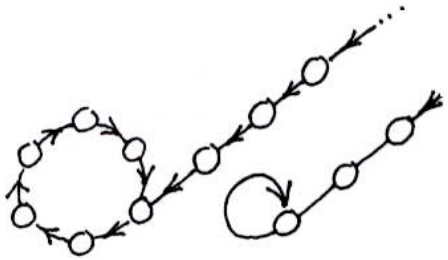
Artificial EvoDevo

DNA ~ 22000 – 25000 genes
Human body ~3.72x10^{13} cells

Artificial Development

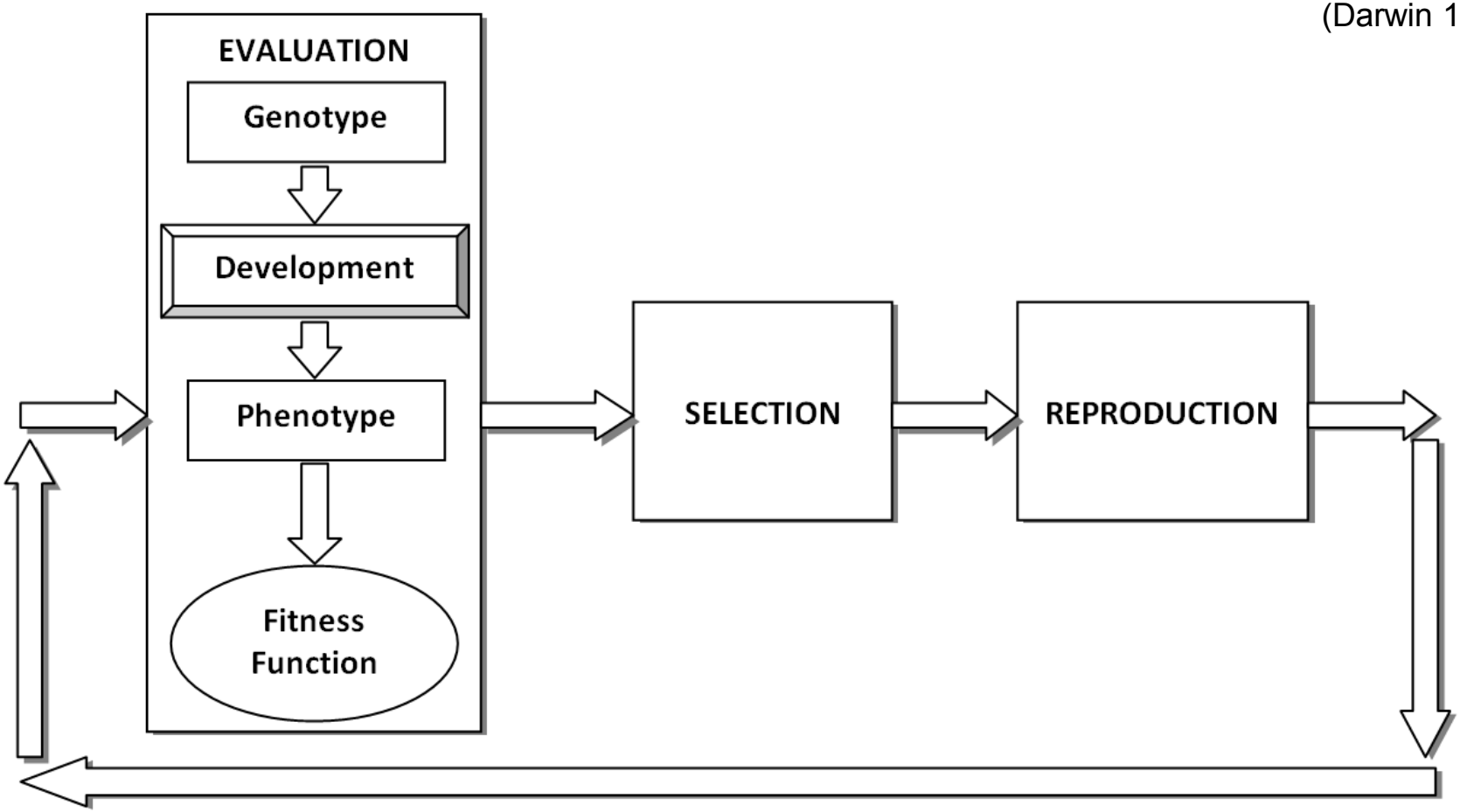






EvoDevo

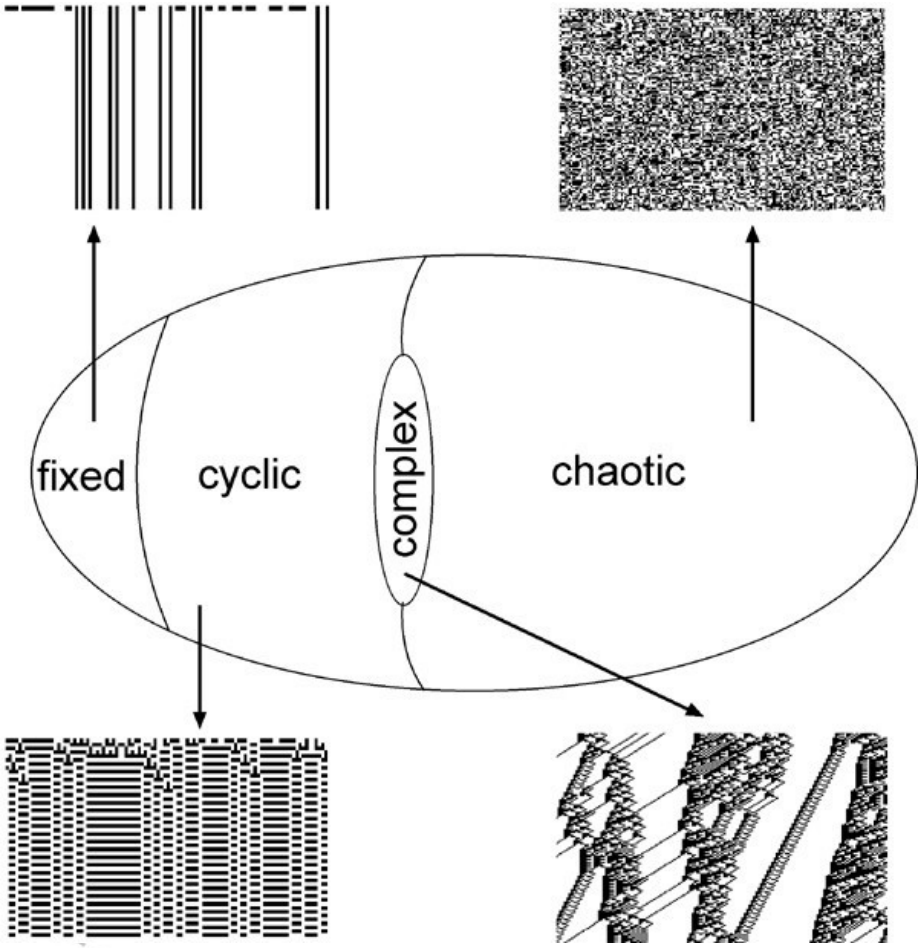
(Darwin 1859)



EvoDevo systems - CA

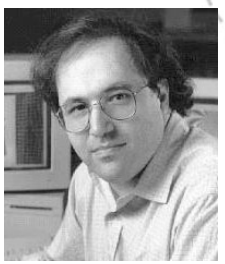
- Cellular Automata can be considered as developmental systems
- Organisms can develop (e.g. grow) from a zygote to a multi-cellular organism (phenotype) according to specific local rules, represented by a genome (genotype)
- The genome specifications and the gene regulatory information control the cells' growth and differentiation
- The behavior of the CA is represented by the emergent phenotype, which is subject to shape and size modification, along the developmental process

Cellular Automata

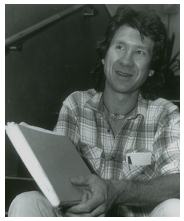


John von Neumann
Precursor of CA,
Universal constructor
Self-reproduction

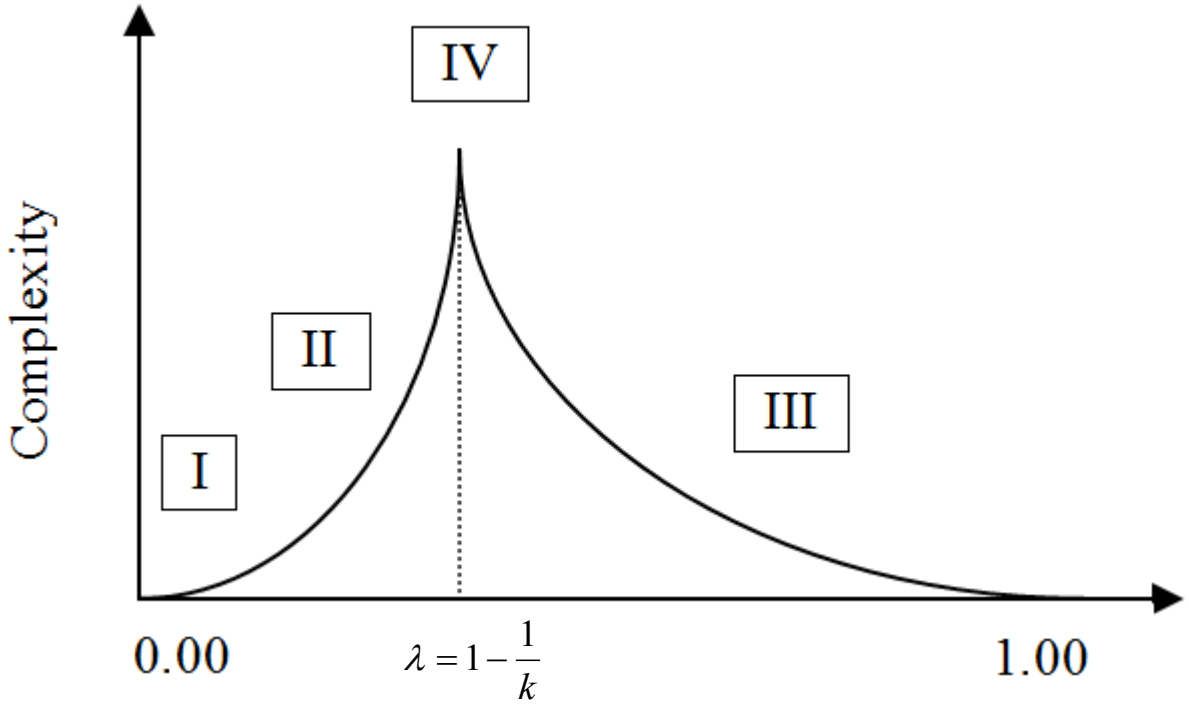
Stephen Wolfram
1D CA classes



Edge of Chaos & Genome Parameters



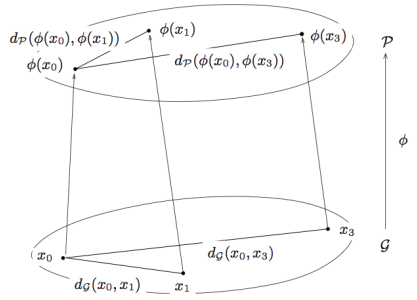
Christopher Langton
Lambda, Alife



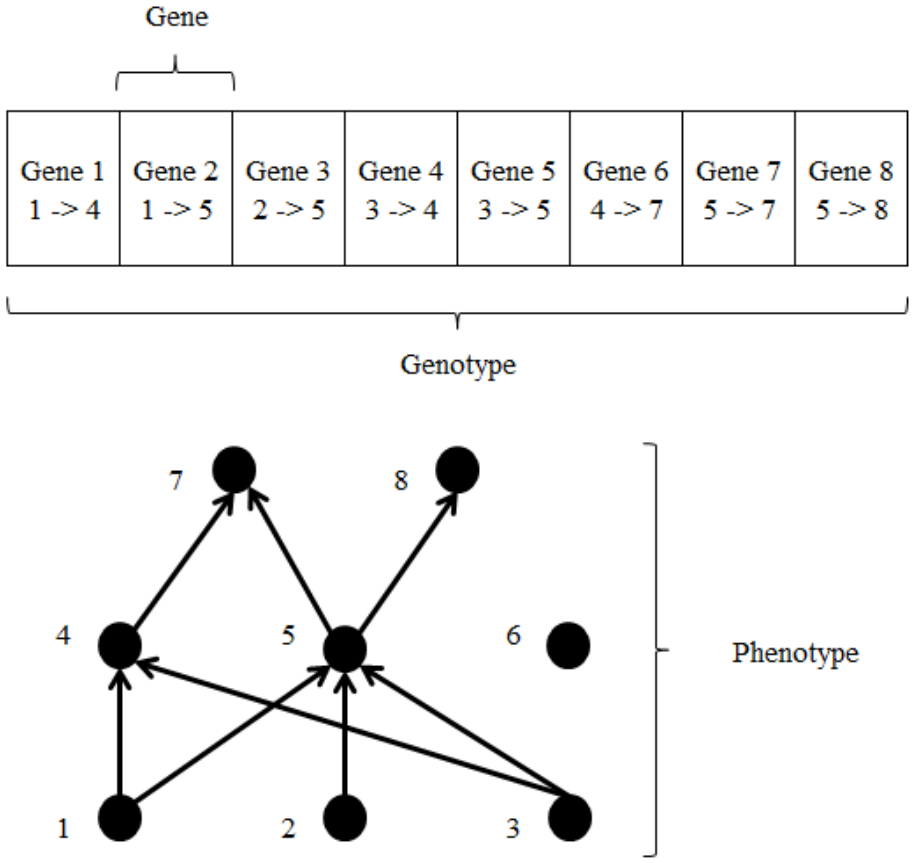
Other parameters:

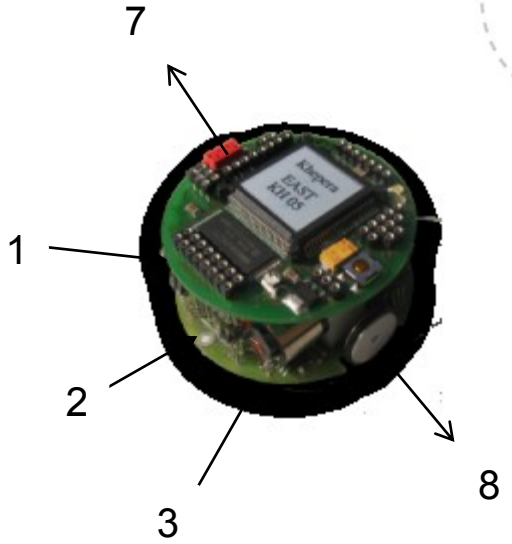
- MFP (Li)
- Sensitivity (Binder)
- Z (Wuensche)
- .....

Genotype-to-Phenotype Encodings



(adapted from "Developmental Mappings and Phenotypic Complexity", Lehre P.C., 2003)





NTNU – Trondheim
Norwegian University of Science and Technology

Genotype-to-Phenotype Encodings

- One-to-one direct Gtype-to-Ptype mapping
- Redundant / many-to-one, neutrality
- Indirect
  - Nature
  - Challenges
    - Restricted set (estimate)
    - If scaled-up?
    - Variable length (speciation)
    - Open ended search

Complexification

The natural and biological process of incremental genome growth and elaboration

- Genomes of different species have different lengths
- LUA (Last Univestal Ancestor): all species diverged from a common ancestor ~ 3.5 - 3.8 billion years ago
- Gene duplication: novelty & potential evolutionary innovation
- Duplicated: redundant but < selection pressure
- 38% of Homo Sapiens genome = gene duplication

- Complexification with direct encodings (Federici & Downing, Stanley & Miikkulainen), e.g. NEAT (NeuroEvolution of Augmenting Topologies)

- Complexification with development

Results

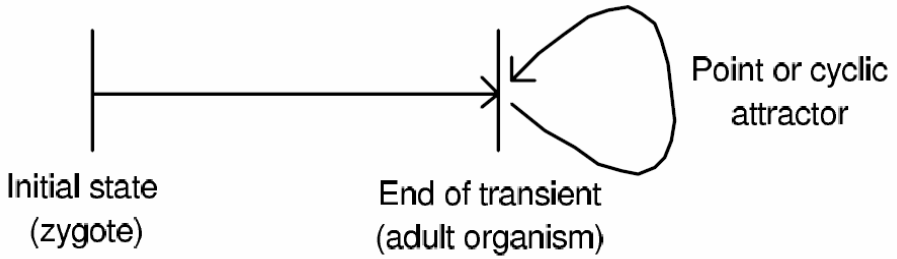
Results Summary

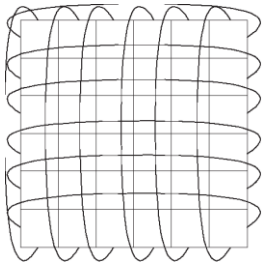
- A.1: not presented
- A.2: in details
- B.1: not presented
- C.1: not presented
- C.2: in details
- D.1: in details
- D.2: shortly (if time allows)
- E.1-E.2-E.3: not included in the thesis

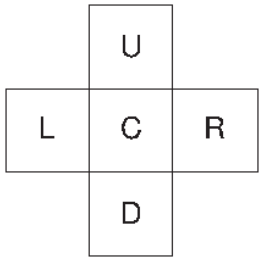
A.2 – Complexity/Evolvability

- Measure genomic properties
- Predict emergent phenotypic properties of artificial organisms
- Genome parameters: λ, M, μ
- How the composition of genome information and gene regulation influences the developmental trajectory

CA model







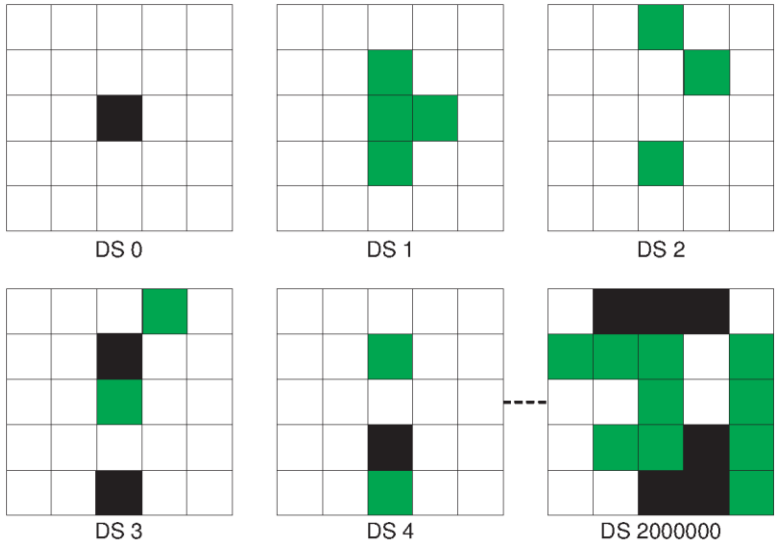
: Void, type 0

: Type 1

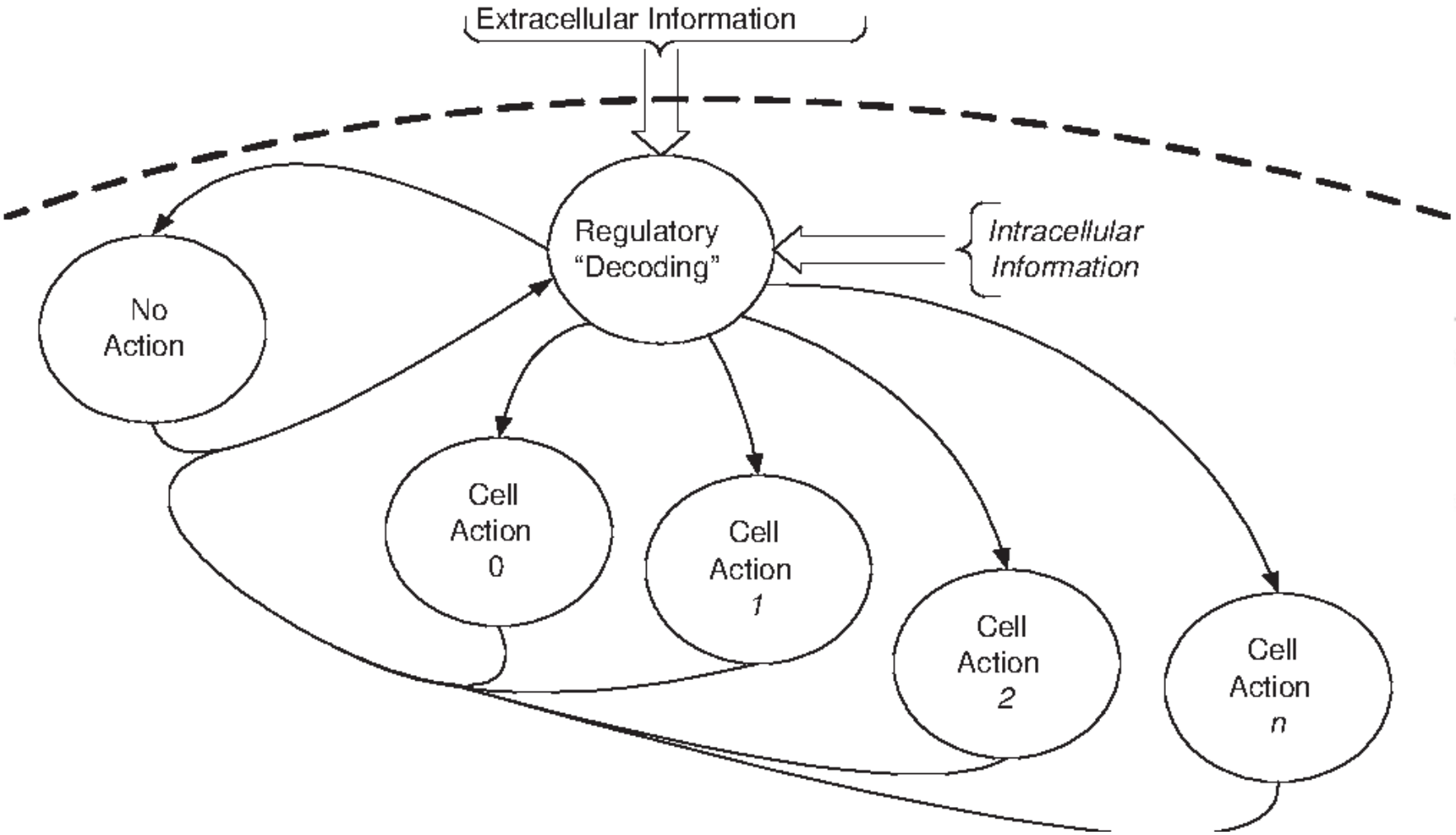
: Type 2

- minimalistic developmental system
- 3 cell types
  (type 0: quiescent, type 1 and type 2 for multicellularity)
- all possible $3^5 = 243$ regulatory input combinations are represented in a development table

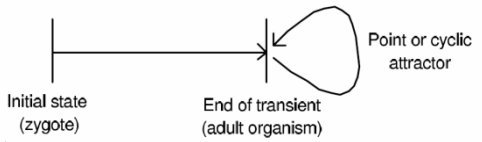
L	R	U	D	C	$C_{(t+1)}$
0	0	0	0	0	0
0	0	0	0	1	{0,1,2}
0	0	0	1	0	{0,1,2}
0	0	0	1	1	{0,1,2}
0	0	1	0	0	{0,1,2}
		:			:
1	1	1	1	1	{0,1,2}
0	0	0	0	2	{0,1,2}
0	0	0	2	0	{0,1,2}
0	0	0	2	1	{0,1,2}
0	0	0	2	2	{0,1,2}
		:			:
2	2	2	2	2	{0,1,2}



NTNU – Trondheim
Norwegian University of Science and Technology


Extracellular Information
Regulatory
"Decoding"
Intracellular
Information
No
Action
Cell
Action
0
Cell
Action
1
Cell
Action
2
Cell
Action
n

Measurements of the Phenotypic Behavior



- trajectory and attractor length: may indicate information about structural and adaptive properties of the organism
  - does development create a stable organism (point attractor) or does the organism end with a self-reorganizing structure that changes form in a cyclic manner (cyclic attractor)?

- growth and change rate: may give information on the activity (internal properties) of the developmental processes
  - growth phase: the organism expand in size toward an ”adult” form
  - change phase: changes in the adult organism (measurement of the adult life of the organism)

Genome Parameters

Evaluation of the genetic information

L	R	U	D	C	$C_{(t+1)}$
0	0	0	0	0	0
0	0	0	0	1	{0,1,2}
0	0	0	1	0	{0,1,2}
0	0	0	1	1	{0,1,2}
0	0	1	0	0	{0,1,2}
		:			:
1	1	1	1	1	{0,1,2}
0	0	0	0	2	{0,1,2}
0	0	0	2	0	{0,1,2}
0	0	0	2	1	{0,1,2}
0	0	0	2	2	{0,1,2}
		:			:
2	2	2	2	2	{0,1,2}

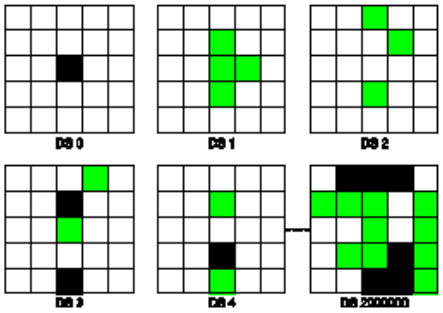
- λ (Lambda): purely regulatory output
- M (Majority): regulatory input and relative output, each entry considered independently
- μ (Sensitivity): overall parameter calculated out of genetic dependency properties

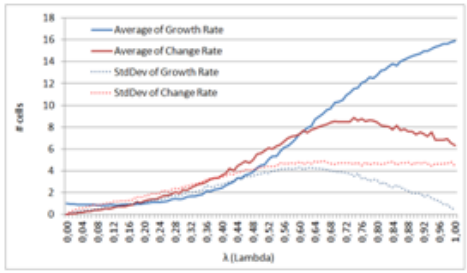
Experimental Setup

Genome Generator → Development Process → Measured Behavior

L	R	U	D	C	C_{next}
0	0	0	0	0	0
0	0	0	0	1	{0,1,2}
0	0	0	1	0	{0,1,2}
0	0	0	1	1	{0,1,2}
0	0	1	0	0	{0,1,2}
		:			:
1	1	1	1	1	{0,1,2}
0	0	0	0	2	{0,1,2}
0	0	0	2	0	{0,1,2}
0	0	0	2	1	{0,1,2}
0	0	0	2	2	{0,1,2}
		:			:
2	2	2	2	2	{0,1,2}

λ
M
μ





State space:

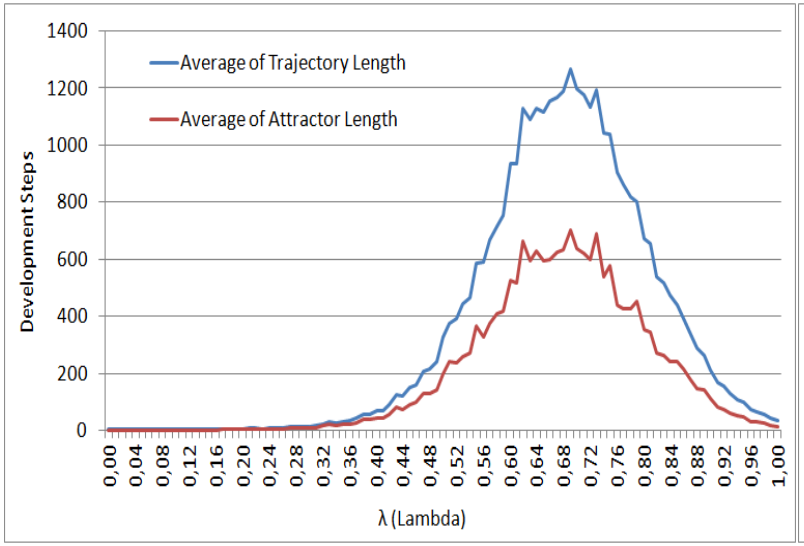
3by3 = 3^9 = 19.683

4by4 = 3^16 = 43.046.721

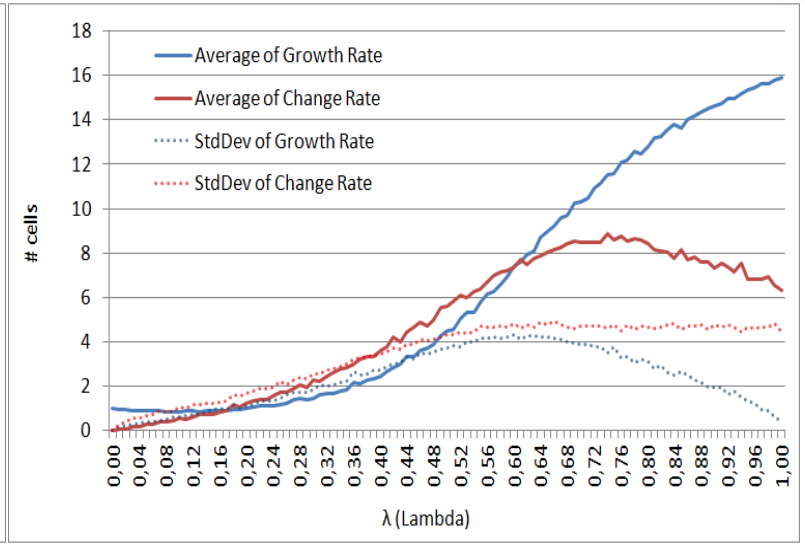
5by5 = 3^25 = 847.288.609.443

Results - λ

Measurements in correlation to λ, average over 1000 tests for each λ value



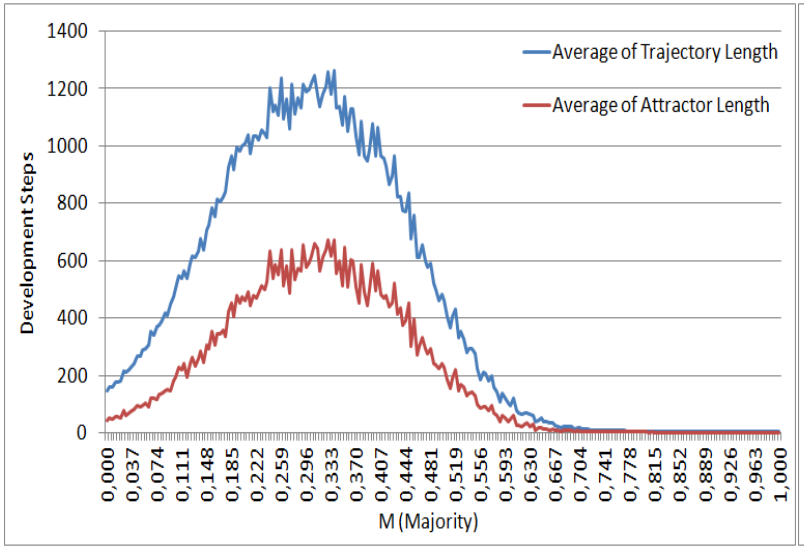
Average trajectory and attractor length



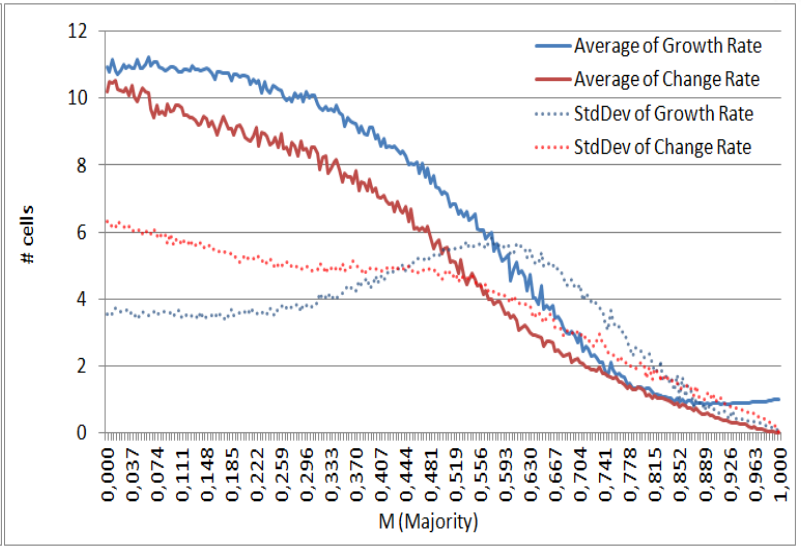
Average growth and change rate

Results - M

Measurements in correlation to M, average over 1000 tests for each M value



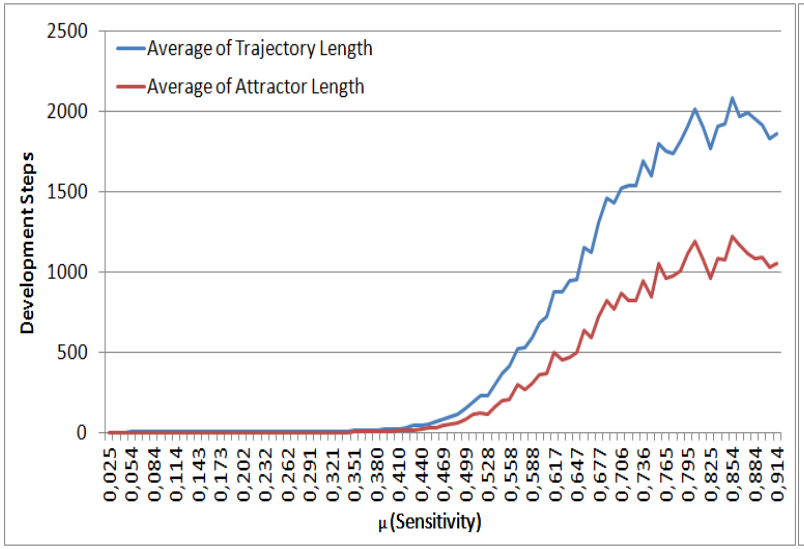
Average trajectory and attractor length



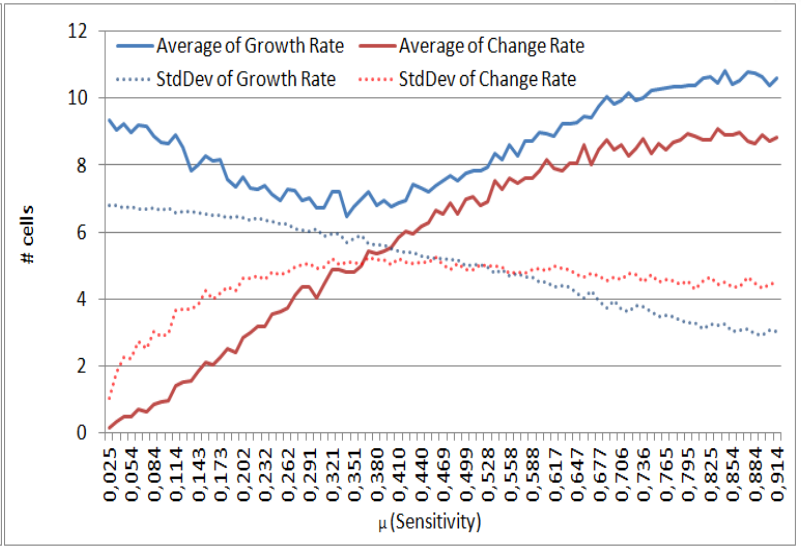
Average growth and change rate

Results - μ

Measurements in correlation to μ, average over 1000 tests for each μ value



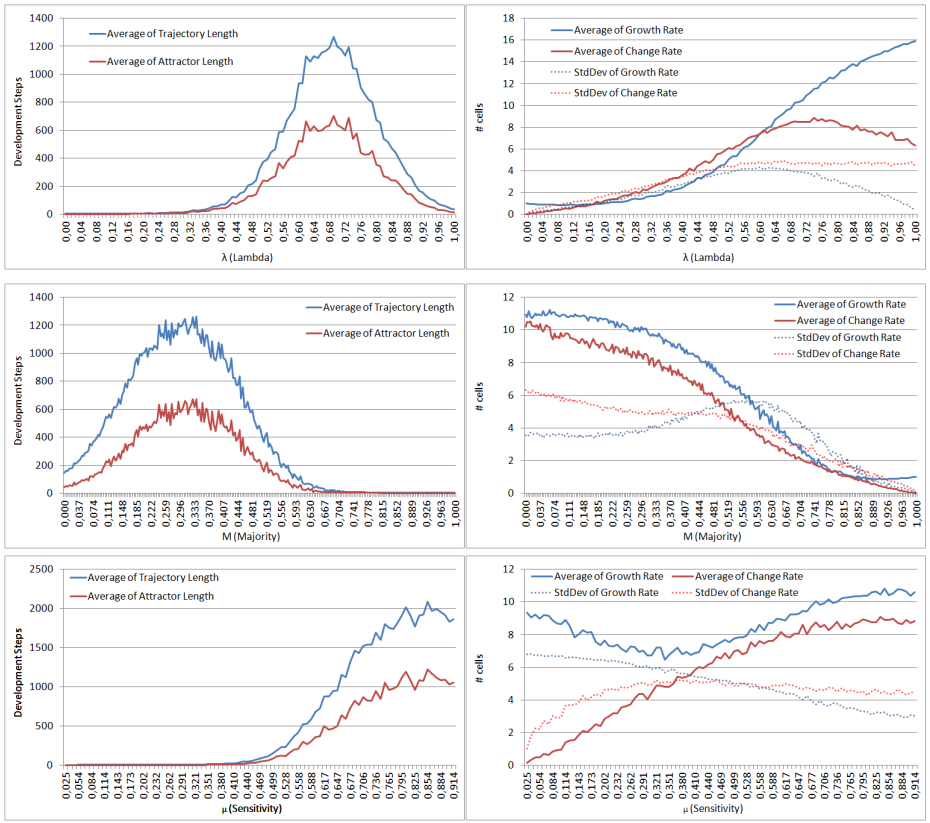
Average trajectory and attractor length



Average growth and change rate

NTNU – Trondheim
Norwegian University of Science and Technology

Comparison


Average of Trajectory Length
Average of Attractor Length
Development Steps
λ (Lambda)
Average of Growth Rate
Average of Change Rate
StdDev of Growth Rate
StdDev of Change Rate
cells
M (Majority)
μ (Sensitivity)

Conclusion A.2

- Parameters as measurement of genomic composition
- Predict developmental behavior
- Relation between genomic composition and developmental properties
- Each genome parameter has a specific ability to measure properties of the resulting organism
- Knowledge of probable developing properties may be helpful at the design stage of an EvoDevo system, if information on the desired target phenotype is known
- Possible to use more parameters together to compose desired developmental behaviors, not achevable with a single parameter

C.2 Evolvability

Goal: genome information (parameters) to guide evolution

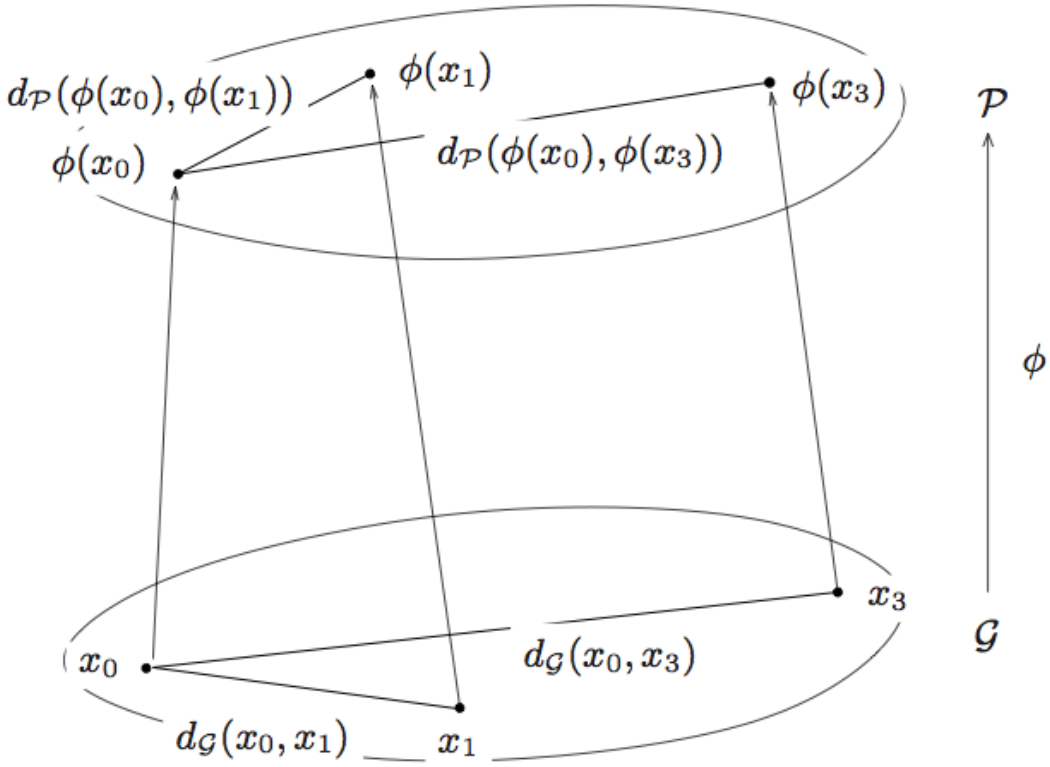
Nature: evolved robust genomes

Robustness VS Evolvability

- Robust: no change in functionality after mutation
- Evolvable: genetic variation, adaptive evolution

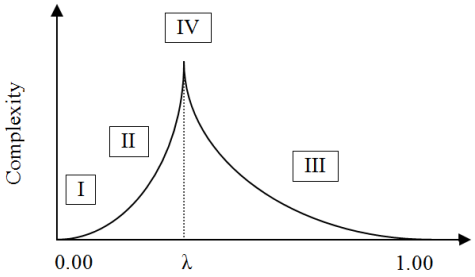
EA: sensitive to mutations

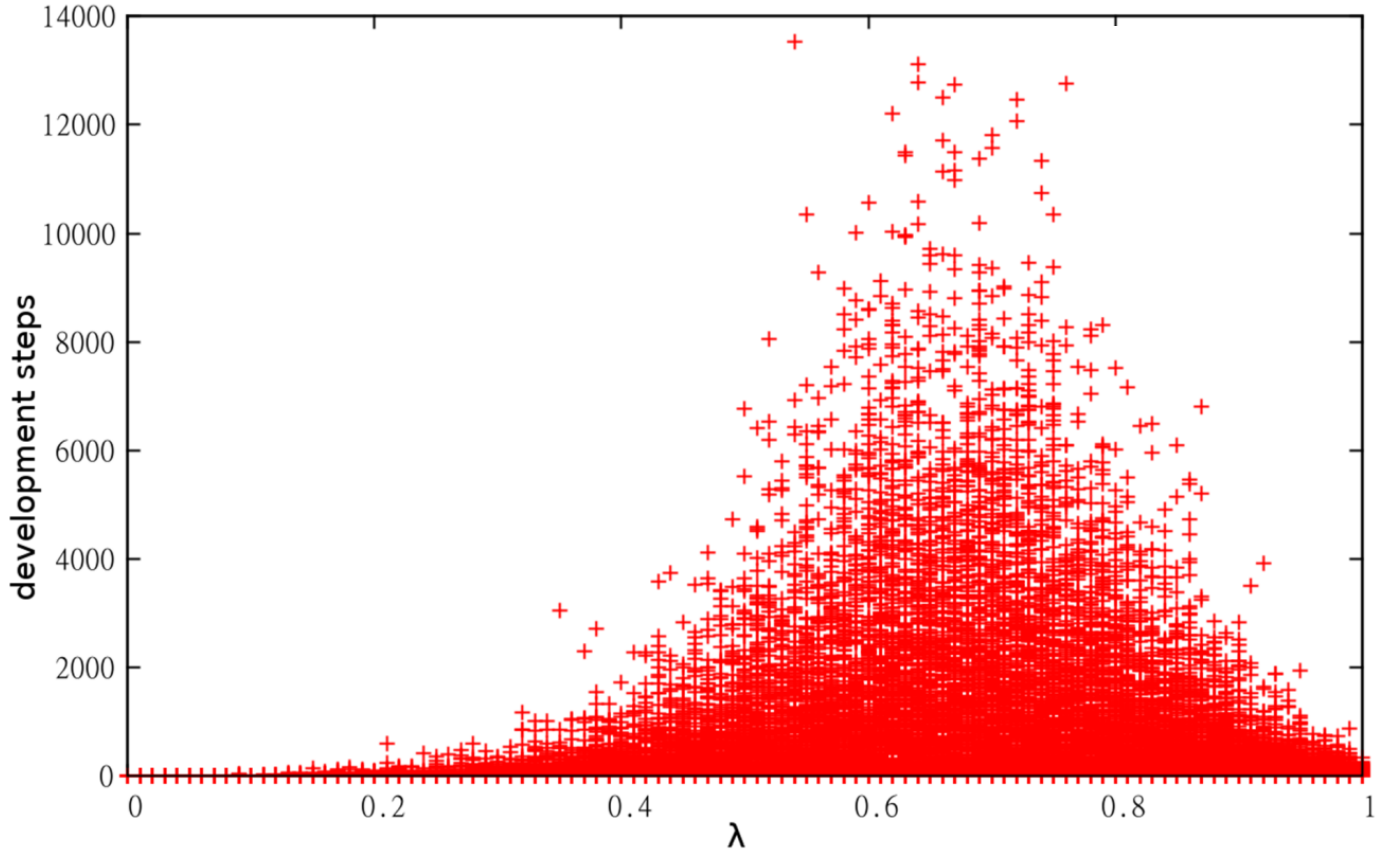
Genotype & phenotype distance



$<d_P=<d_G$

(adapted from “Developmental Mappings and Phenotypic Complexity”, Lehre P.C., 2003)





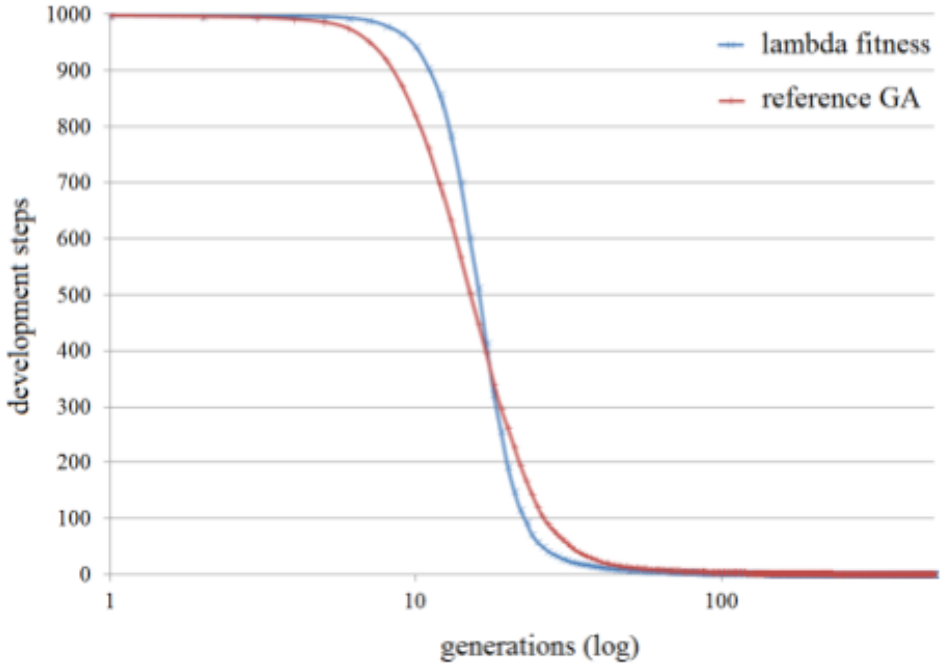
eim
Norwegian University of Science and Technology

Experimental Setup

- Genetic Algorithm (details in paper)
- Initial population = most unfit (all transitions to quiescent state)
- Standard fitness VS parameter in fitness

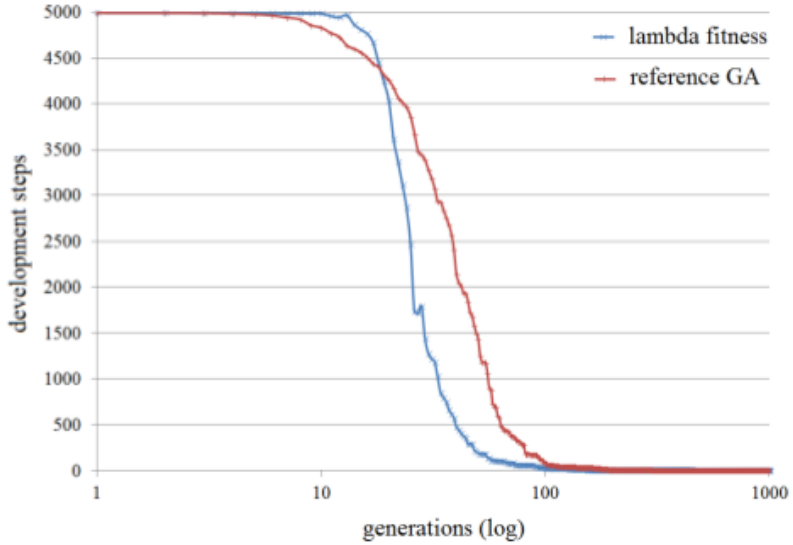
$$CFitness = Fitness + Fitness \times \frac{Abs(HiLambda - Lambda)}{HiLambda} \times ratio$$

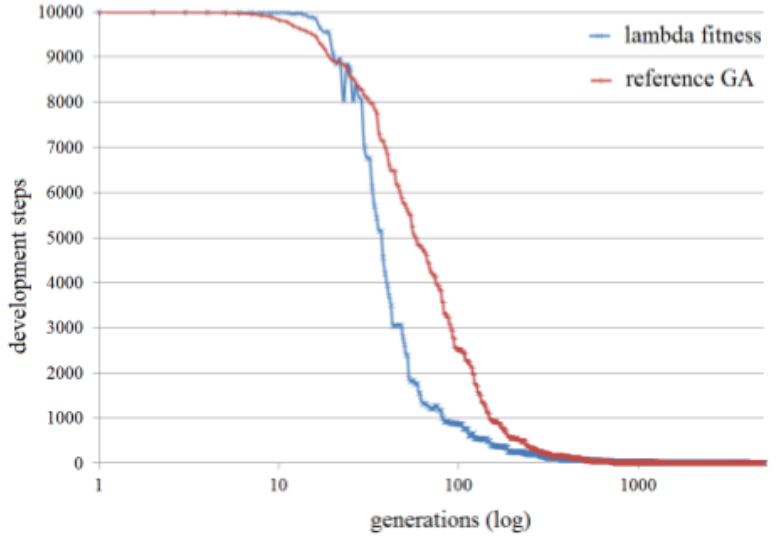
Results - target 1000 dev.steps

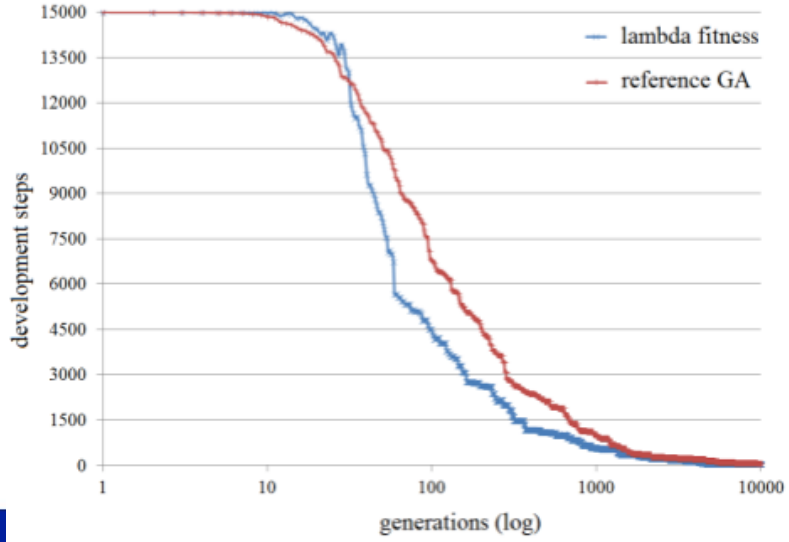


GA	Void genomes (plotted)		Randomized genomes	
Reference GA	443,60	-	372,00	-
Lambda fitness	365,62	-17,57%	351,33	-5,56%

5000, 10000, 15000







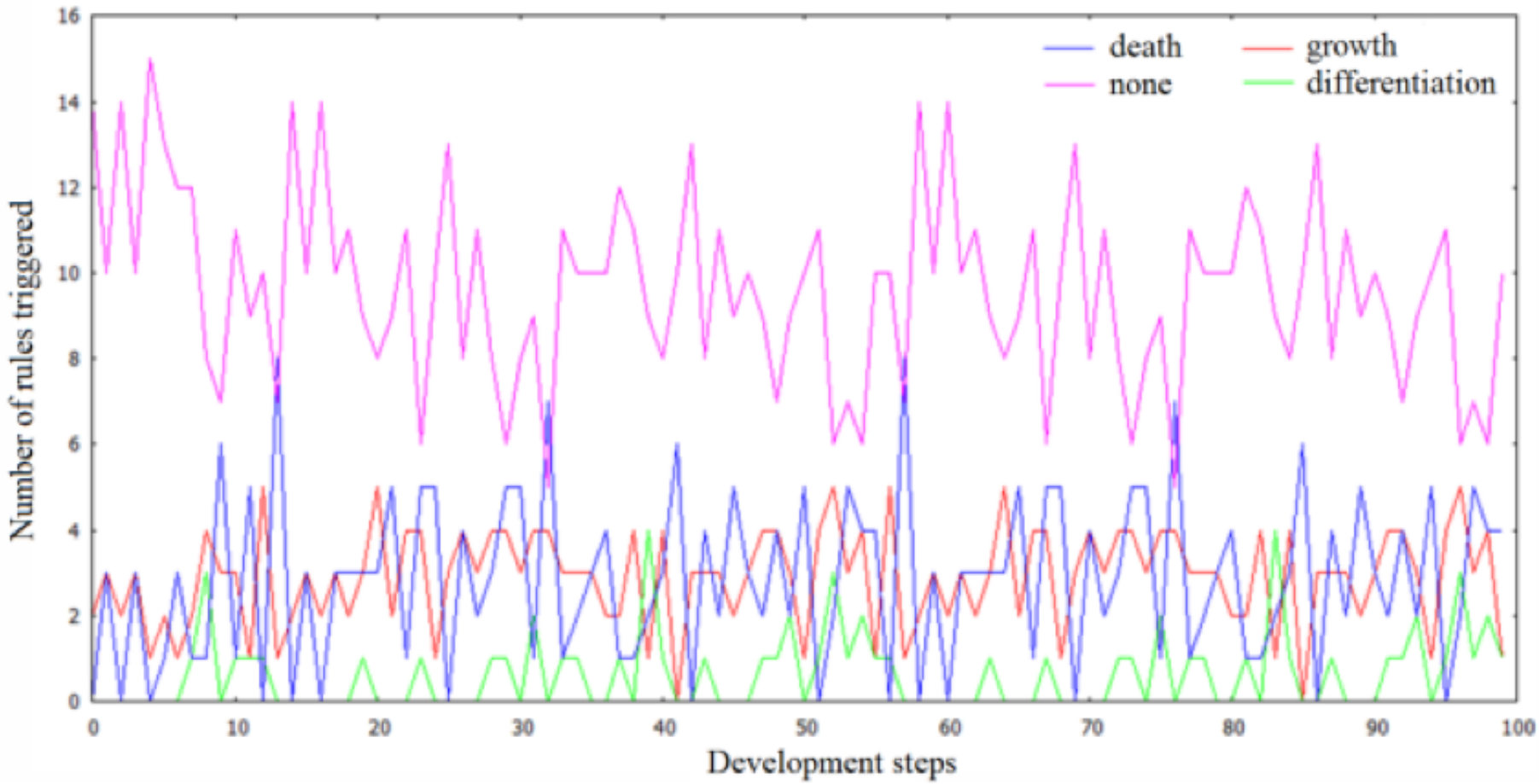
Sub-Transitions

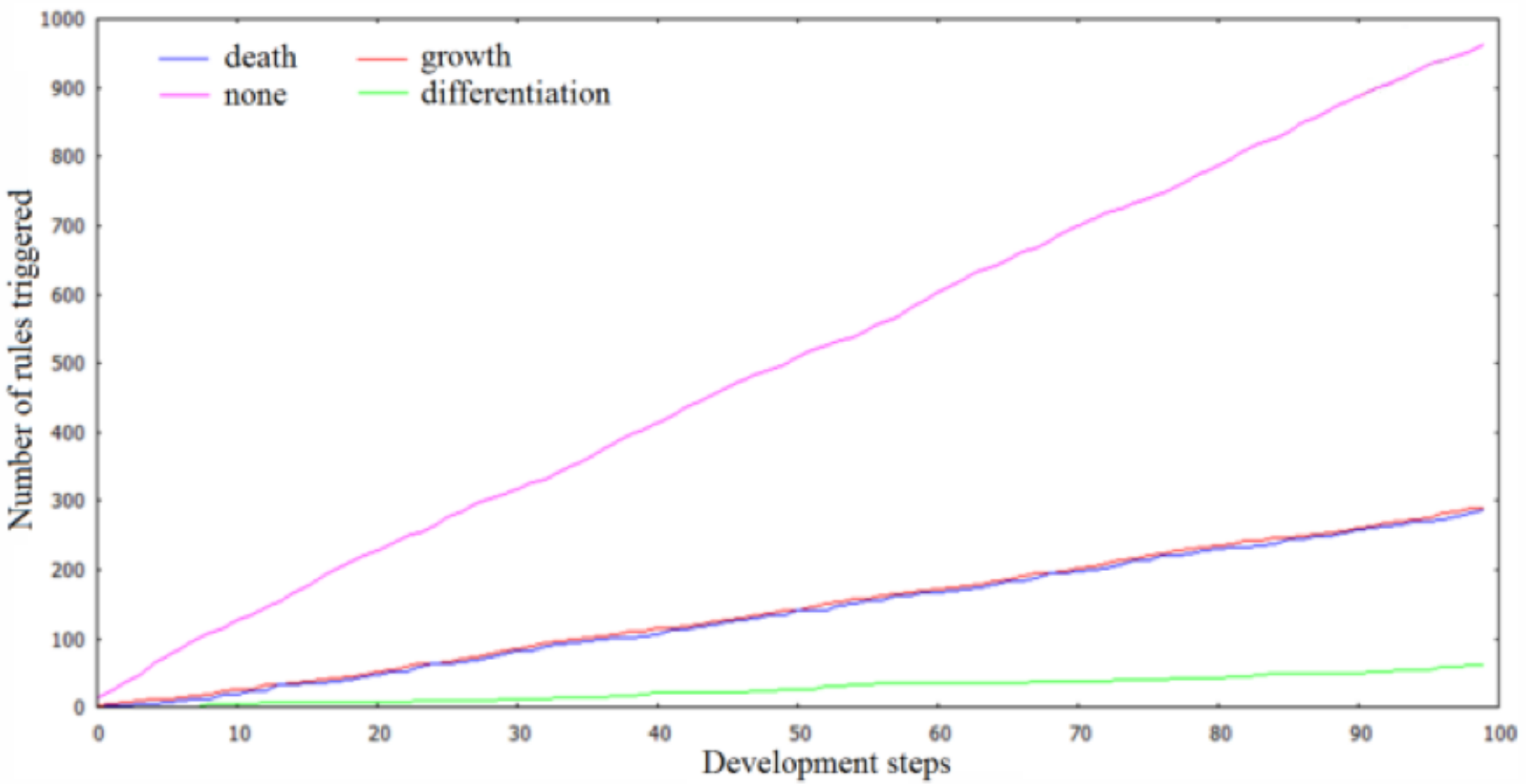
L	R	U	D	C	$C_{(t+1)}$
0	0	0	0	0	0
0	0	0	0	1	{0,1,2}
0	0	0	1	0	{0,1,2}
0	0	0	1	1	{0,1,2}
0	0	1	0	0	{0,1,2}
		:			:
1	1	1	1	1	{0,1,2}
0	0	0	0	2	{0,1,2}
0	0	0	2	0	{0,1,2}
0	0	0	2	1	{0,1,2}
0	0	0	2	2	{0,1,2}
		:			:
2	2	2	2	2	{0,1,2}

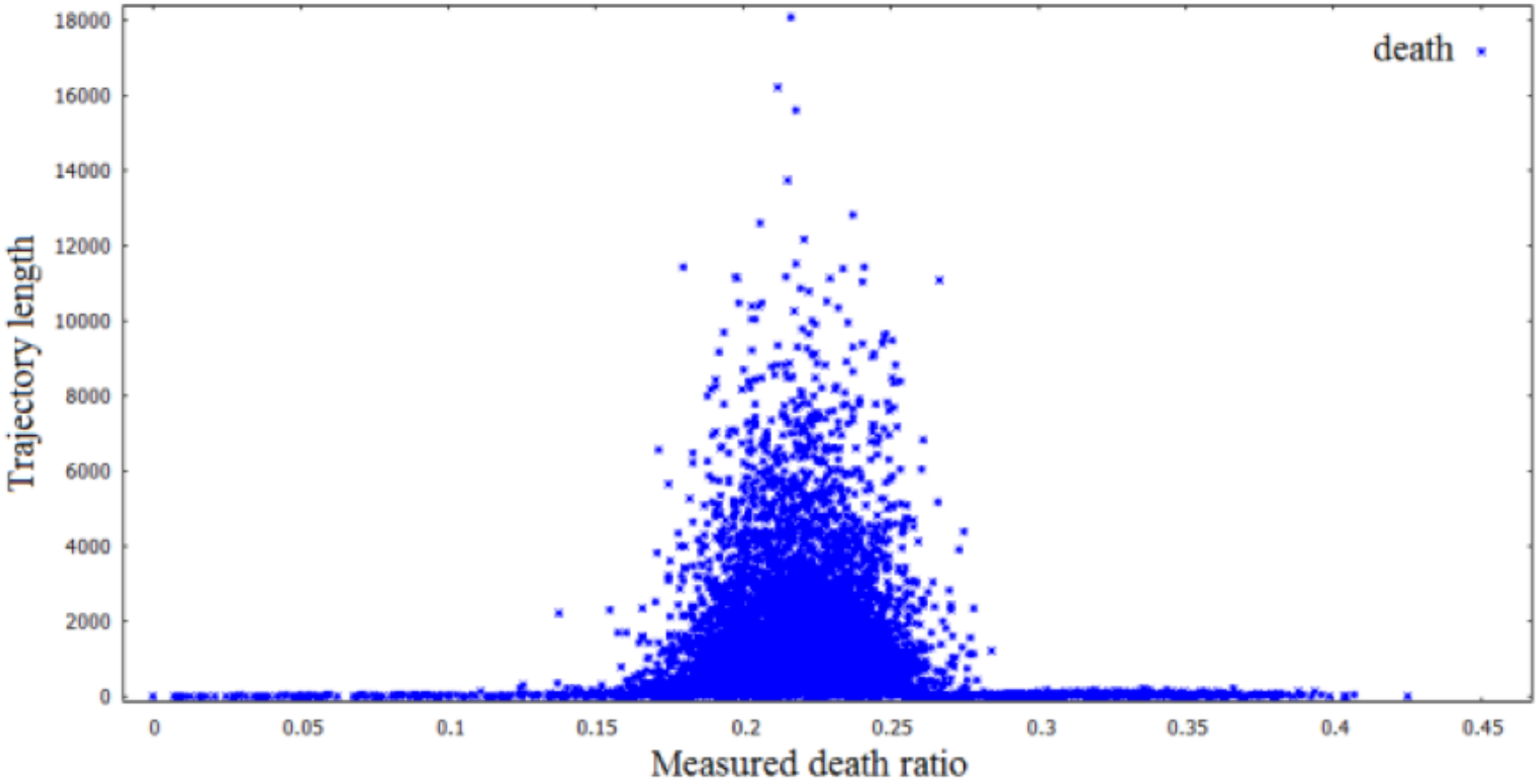
- Growth
- Differentiation
- Death (Lambda – quiescent state)
- No-Change

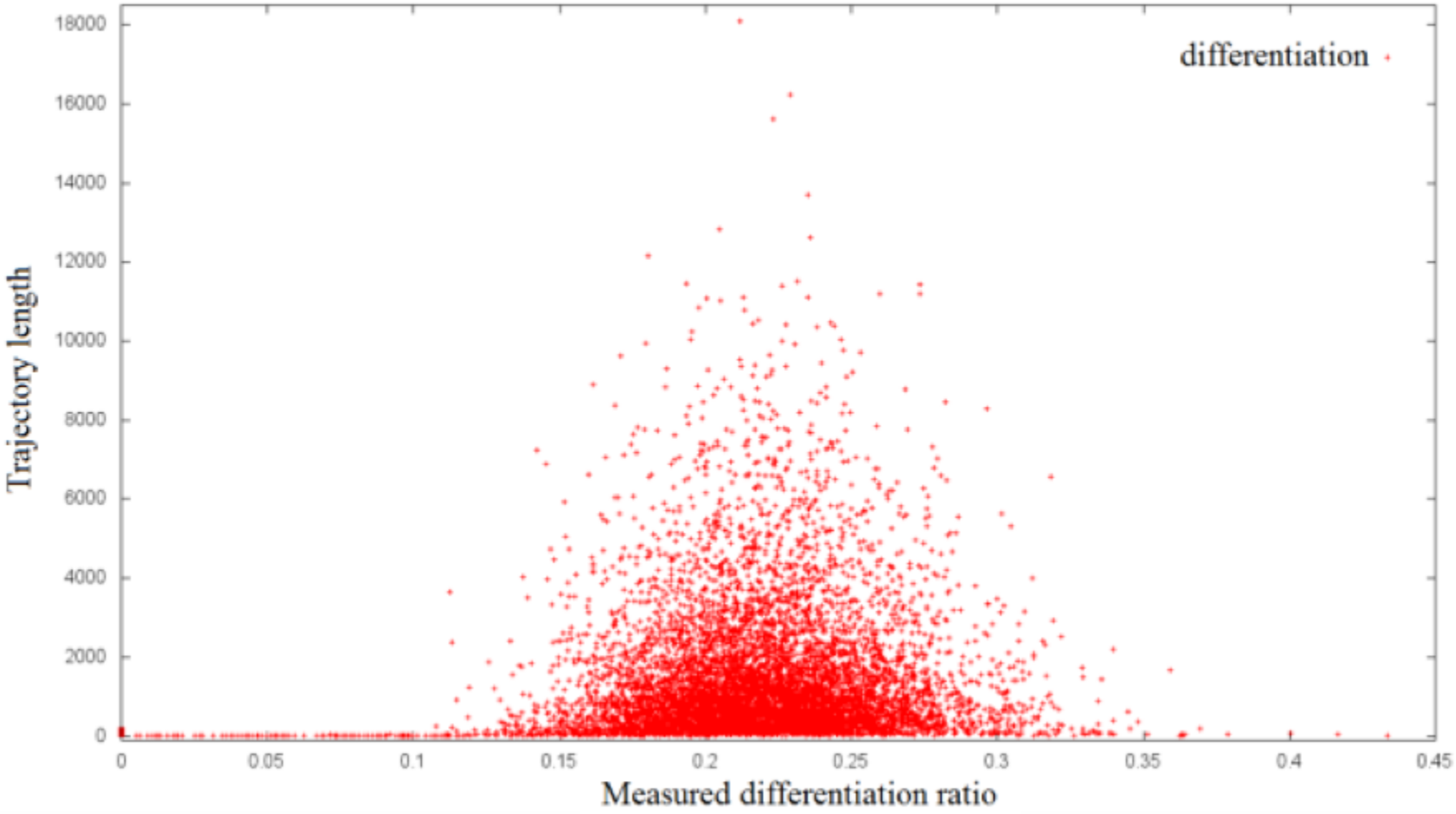
Lambda: single sub-transition parameter

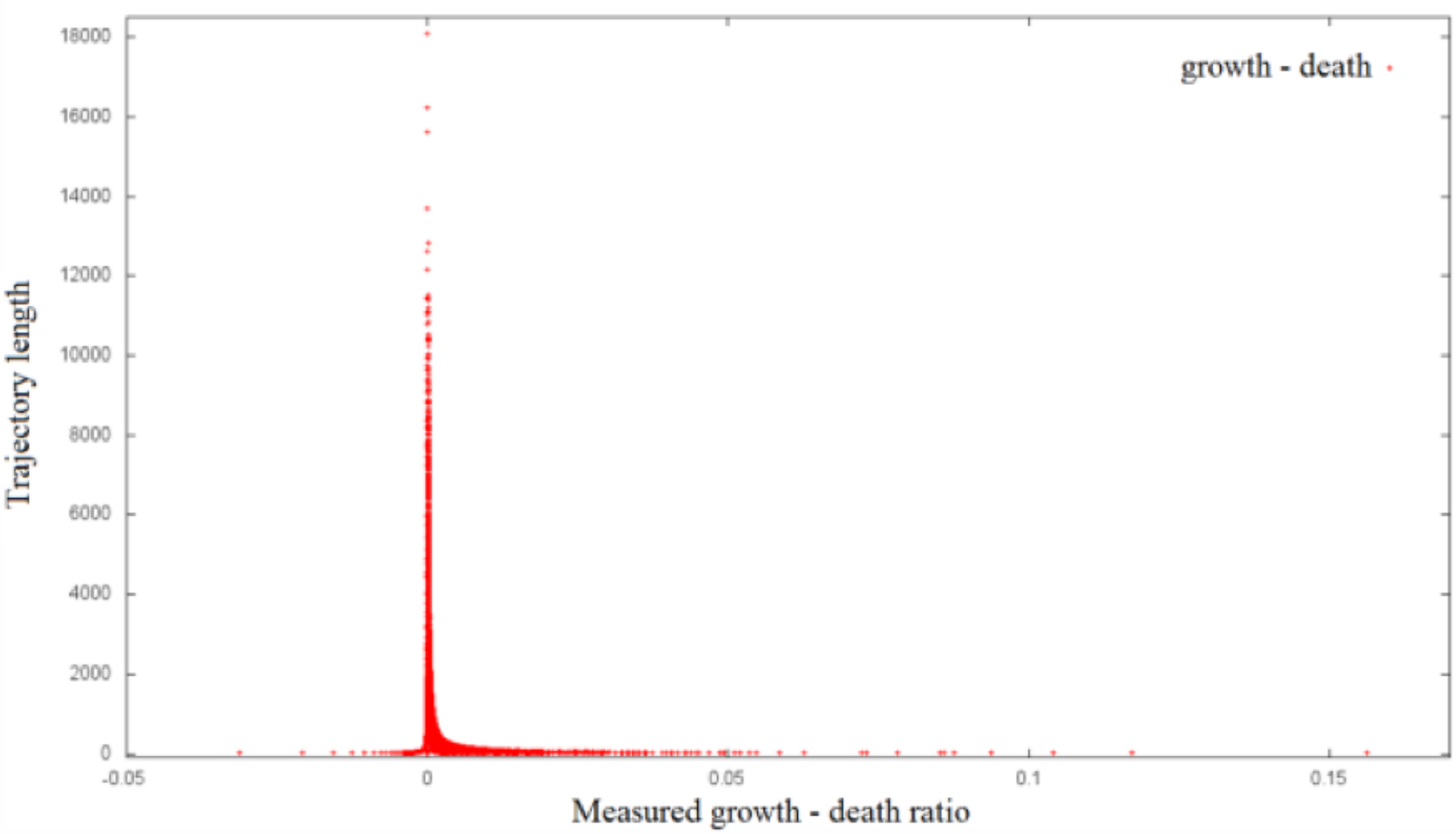
More states = more sub-transition classes, Lambda less meaning, possible to build custom parameter

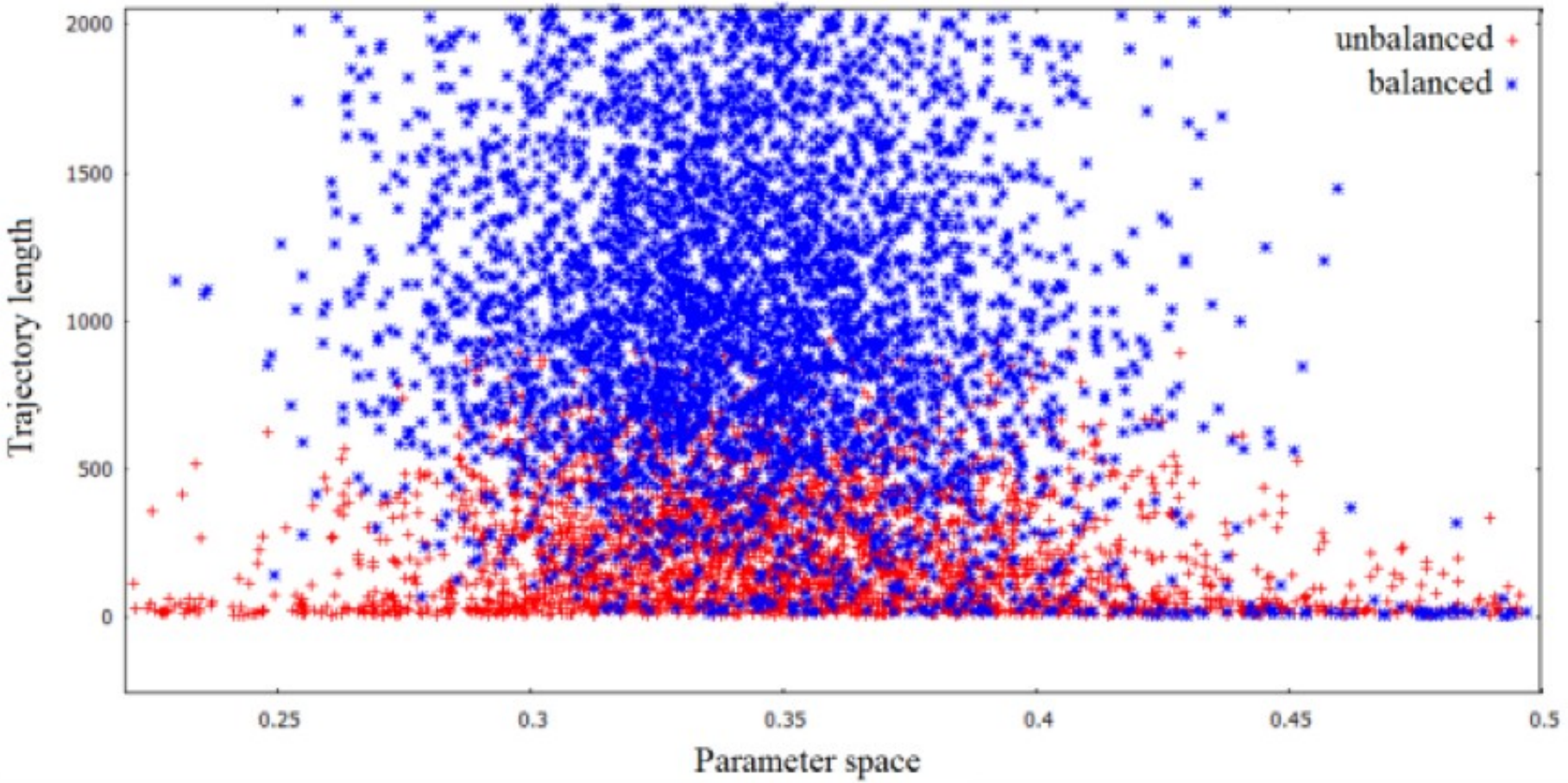

death
growth
none
differentiation
Number of rules triggered
Development steps
0
2
4
6
8
10
12
14
16
0
10
20
30
40
50
60
70
80
90
100


death
growth
none
differentiation
Number of rules triggered
Development steps
0
100
200
300
400
500
600
700
800
900
1000
0
10
20
30
40
50
60
70
80
90
100


death
Trajectory length
Measured death ratio
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45


differentiation
Trajectory length
Measured differentiation ratio
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45


growth - death
Trajectory length
Measured growth - death ratio
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
-0.05
0
0.05
0.1
0.15


unbalanced
balanced
Trajectory length
2000
1500
1000
500
0
0.25
0.3
0.35
0.4
0.45
0.5
Parameter space

Conclusion C.2

- Genome information to guide evolution
- Vast search space, indirect G-Ptype mapping, development
- Where the target behavior is more likely to be found
- Lambda in fitness to speedup convergence
- Sub-Transitions
- Composite parameters
- Growth – Death transition
- Filter

D.1 Scalability

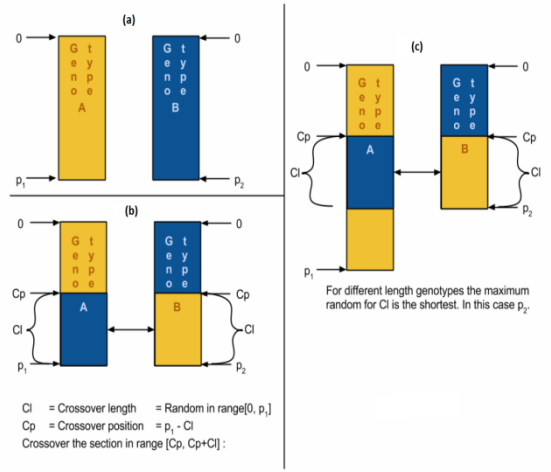
Different Developmental Model
- Why?
- How?

Motivation (Why):

- In nature genomes of different species have different lengths
- Scaling of artificial systems (state, search and solution space)
- Genotype representation problem (estimated, heuristics)

Genome Growth (How)

- Allows speciation
- Through gene duplication (in nature)
- Complexification (incremental elaboration)
- Compare full vs restricted vs growing (genomes)

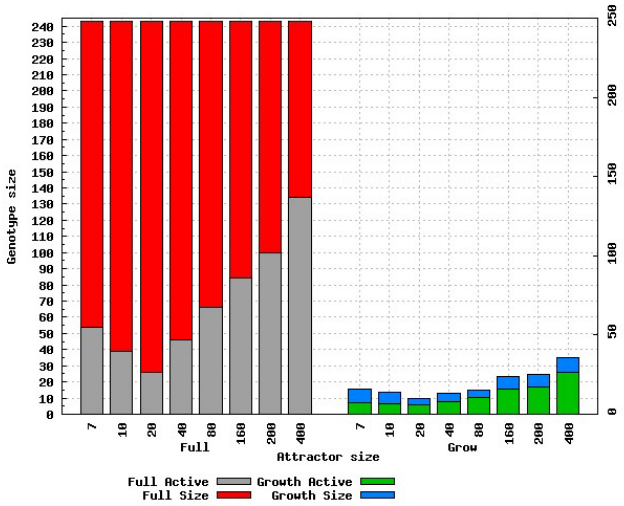


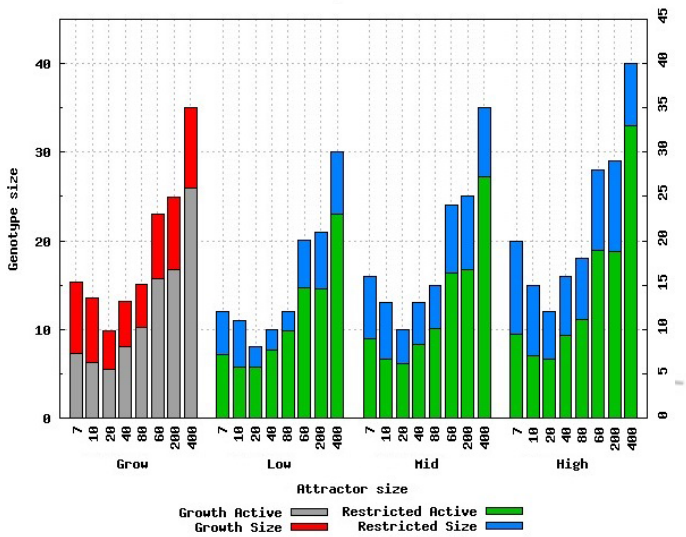
Regulation mechanisms:

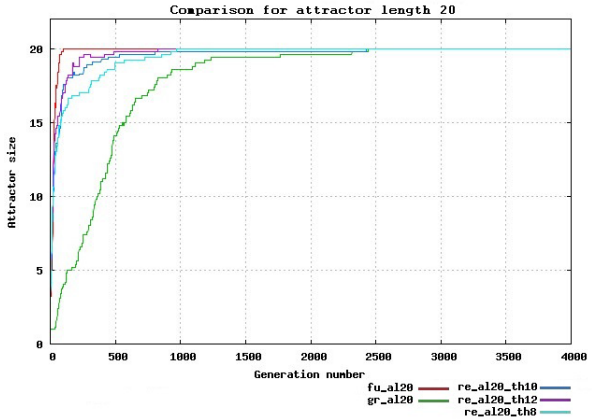
- Upper bound, duplication rate, optimization time, elitism

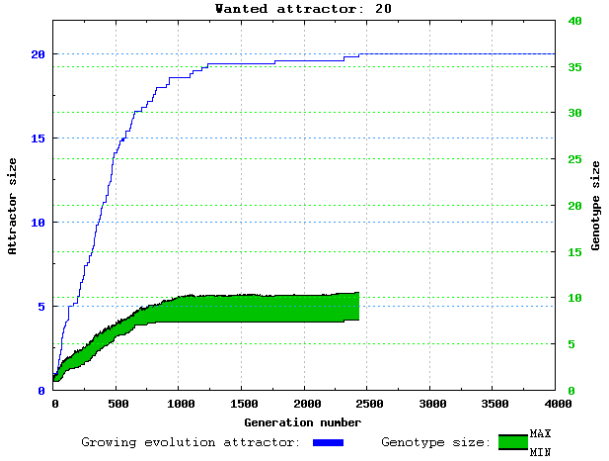
Selection:

- Σ (actual fitness, exploitation parameter, innovation parameter)



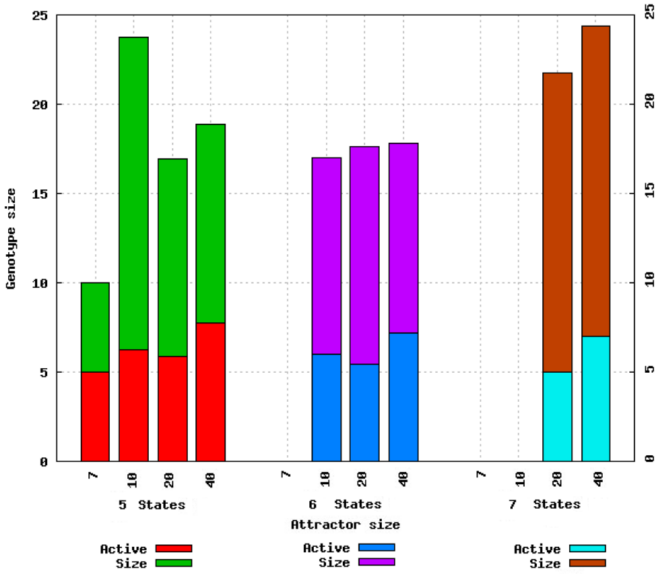


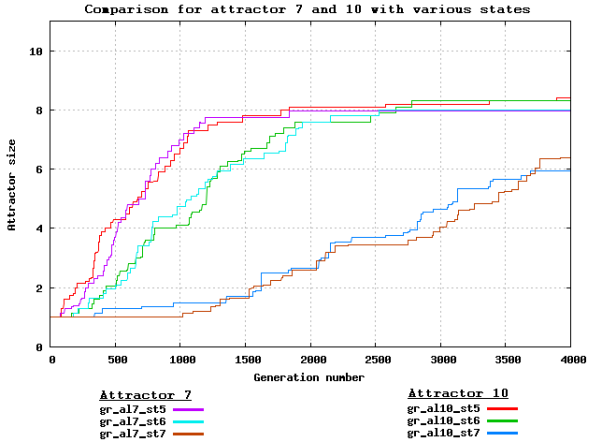


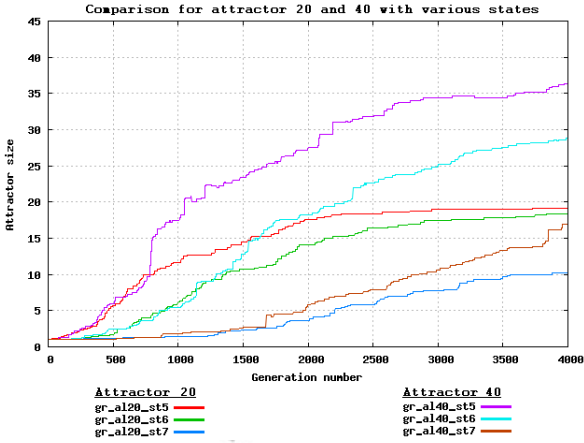


Scalability in search space – genome comparison

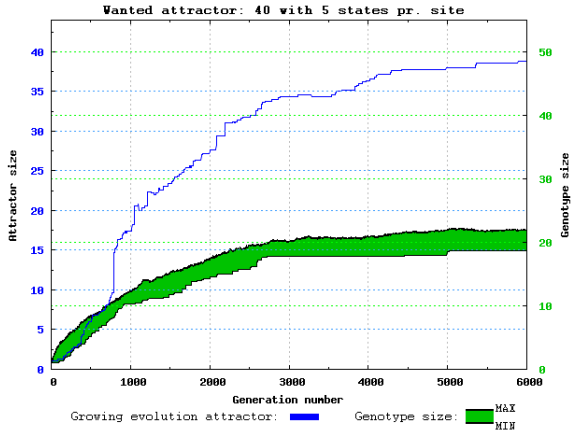
NTNU – Trondheim
Norwegian University of Science and Technology

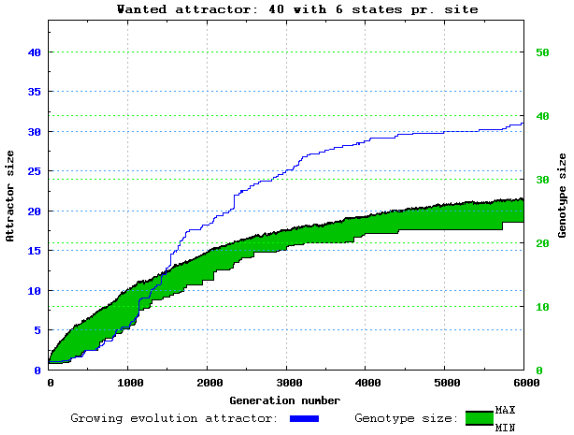


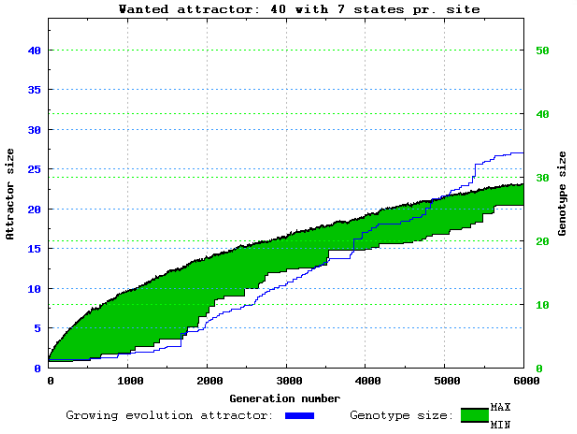




Scalability in state space

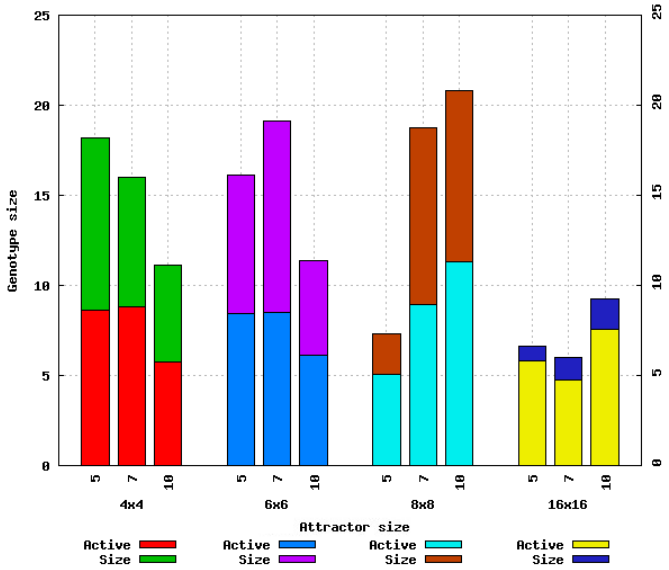


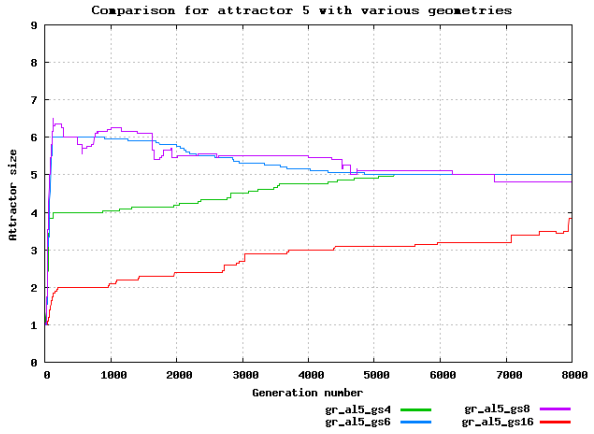


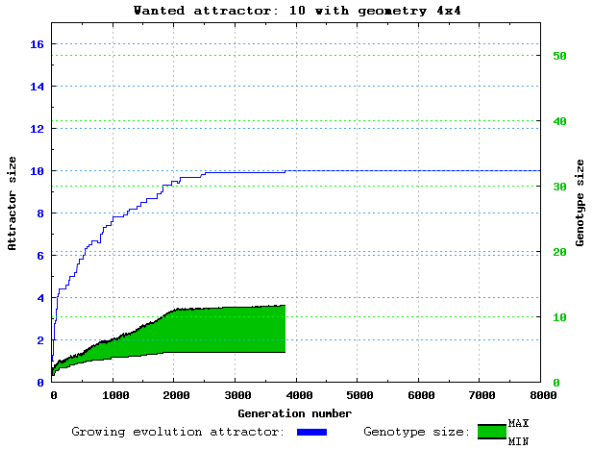


NTNU – Trondheim
Norwegian University of Science and Technology

Scalability in solution space - geometry





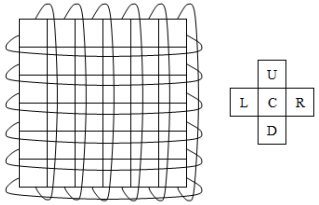


Conclusion D.1

- Evolutionary growth of genome representations
- Compact and effective genomes
- Scalability of search space
- Scalability of state space
- Scalability of phenotypic resources
- Start in low dimensional space
- Incrementally increase genotype complexity

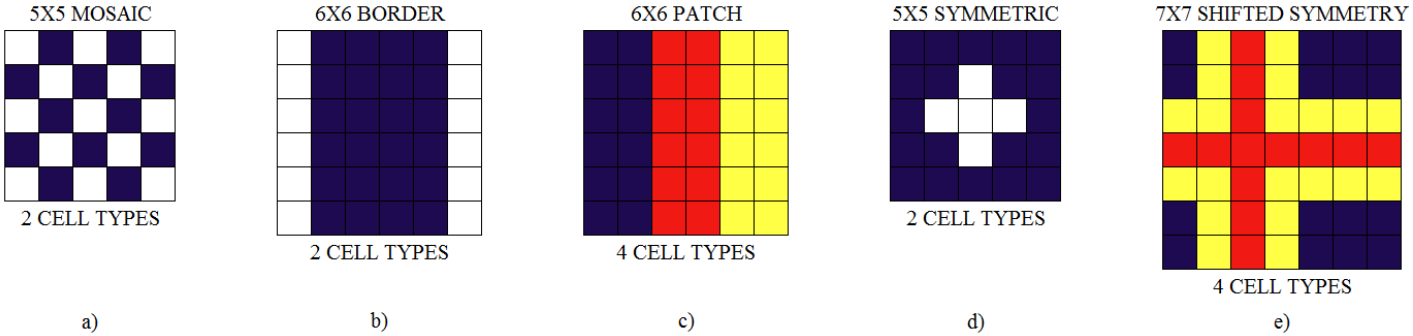
D.2

- Genome Growth
- Instruction-Based Development



L	R	U	D	C	C(t+1)
0	0	0	0	0	0
0	0	0	0	1	{0,1,2, ..., n}
0	0	0	1	0	{0,1,2, ..., n}
0	0	0	1	1	{0,1,2, ..., n}
0	0	1	0	0	{0,1,2, ..., n}
:	:	:	:	:	:
1	1	1	1	1	{0,1,2, ..., n}
0	0	0	0	2	{0,1,2, ..., n}
0	0	0	2	0	{0,1,2, ..., n}
0	0	0	2	1	{0,1,2, ..., n}
0	0	0	2	2	{0,1,2, ..., n}
:	:	:	:	:	:
n-1	n-1	n-1	n-1	n-1	{0,1,2, ..., n}
0	0	0	0	n	{0,1,2, ..., n}
0	0	0	n	0	{0,1,2, ..., n}
0	0	0	n	1	{0,1,2, ..., n}
0	0	0	n	2	{0,1,2, ..., n}
:	:	:	:	:	:
n	n	n	n	n	{0,1,2, ..., n}

Instruction	Description	Meaning	Code
AND	$N(op_1) = N(op_1) \wedge N(op_2)$	AND operation	0
OR	$N(op_1) = N(op_1) \vee N(op_2)$	OR operation	1
XOR	$N(op_1) = N(op_1) \oplus N(op_2)$	XOR operation	2
NOT	$N(op_1) = \neg N(op_1)$	NOT operation	3
INV	$N(op_1) = n - N(op_1)$	Inverse state	4
MIN	$N(op_1) = \min(N(op_1), N(op_2))$	Minimum	5
MAX	$N(op_1) = \max(N(op_1), N(op_2))$	Maximum	6
SET	$N(op_1) = N(op_2)$	Set value	7
INC	$N(op_1) = N(op_1) + 1$	Increment	8
DEC	$N(op_1) = N(op_1) - 1$	Decrement	9
SWAP	$N(op_1) \leftrightarrow N(op_2)$	Swap	10
ROR	LCR → RLC	Rotate right	11
ROL	LCR → LCR	Rotate left	12
ROU	UCD → CDU	Rotate up	13
ROD	UCD → DUC	Rotate down	14
NOP	$N(op_1) = N(op_1)$	No operation	15

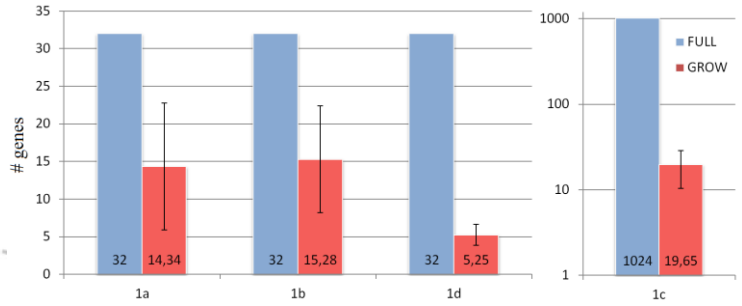


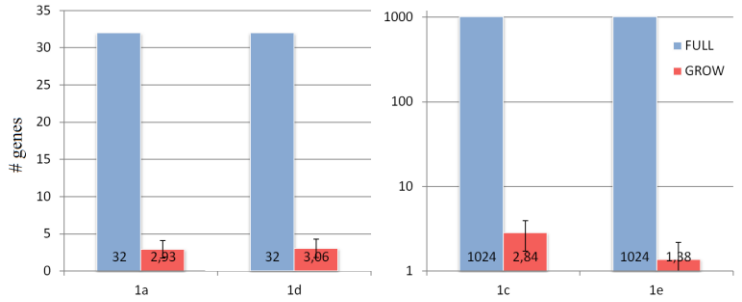
Development

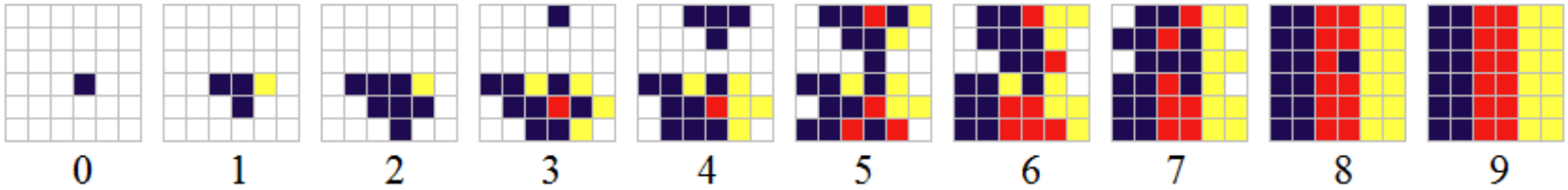
Fig.	Table-based Evolution						
	Success Rate %	Genotype Size (# genes) Max	Avg	Min	StDev	Generations Avg.	StDev.
2a	58	32	32	32	0	1336	2294
2b	69	32	32	32	0	2254	2501
2c	19	1024	1024	1024	0	5002	3157
2d	23	32	32	32	0	2668	2942
Fig.	**Instruction-based Growing Evolution**						
	Success Rate %	Genotype Size (# genes) Max	Avg	Min	StDev	Generations Avg.	StDev.
2a	98	31	14.34	5	8.4318	1257	1152
2b	98	31	15.28	5	7.0973	3956	1690
2c	46	46	19.65	6	9.2236	6424	1922
2d	100	13	5.25	4	1.4097	285	108

Replication

Fig.	Table-based Evolution						
	Success Rate %	Genotype Size (# genes) Max	Avg	Min	StDev	Generations Avg.	StDev.
2a	85	32	32	32	0	775	1393
2c	8	1024	1024	1024	0	4331	3576
2d	1	32	32	32	0	8259	0
2e	0	1024	1024	1024	0	-	-
Fig.	**Instruction-based Growing Evolution**						
	Success Rate %	Genotype Size (# genes) Max	Avg	Min	StDev	Generations Avg.	StDev.
2a	100	7	2.93	2	1.1742	39.7	19.6
2c	100	6	2.84	2	1.1166	39.6	22.3
2d	100	8	3.06	2	1.2128	41.8	20.5
2e	100	5	1.38	1	0.8012	9.4	10.7





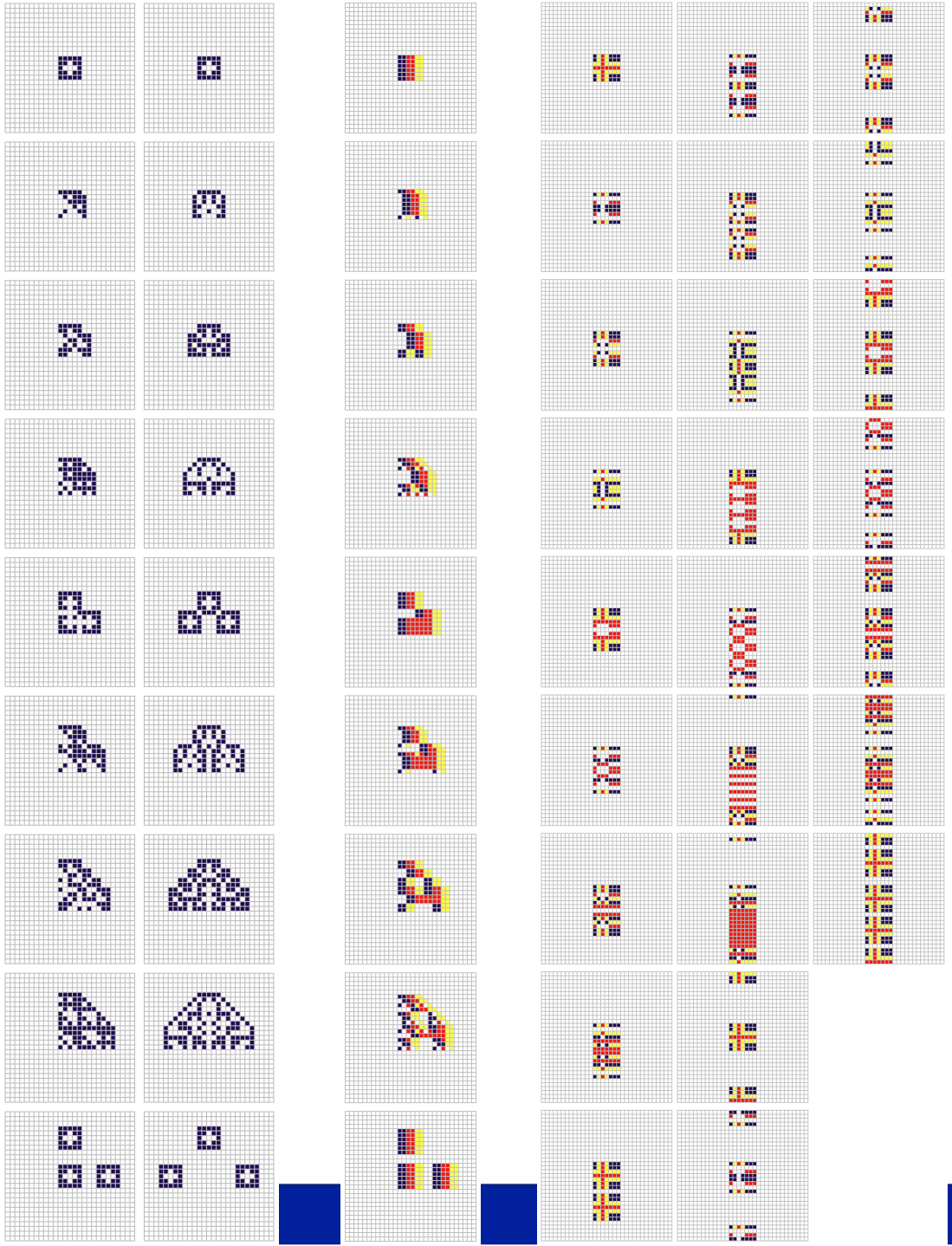


6 1 3	3 3 1	8 3 4	11 4 3	6 3 0	2 4 3	1 1 4	2 1 4	6 4 1	15 4 3	5 2 0	1 2 4	13 4 0	0 2 0

- Example of evolved program for the development of structure 2c – patch structure
- After development step 9 the structure remains stable (point attractor)
- The program is composed by 14 instructions (one instruction each gene)

- INSTRUCTION CODE, OPERAND 1, OPERAND 2 (if the operand is not applicable for the given instruction, the value is ignored)

- Operands: UP = 0, RIGHT = 1, DOWN = 2, LEFT = 3, CENTRE = 4.



Examples of evolved solutions for the replication of the structures 2d, 2c and 2e.

Conclusions and Further Work

Conclusion

- Genome parameters, forecast emergent developmental phenotypes (A.1, A.2)
- Abstract measures of developmental complexity (B.1, A.1, A.2)
- Genome parameters as evolvability evaluation (C.1)
- Genome parameters to guide evolution (C.2)
- Evolutionary growth of genomes (D.1, D.2)

GUIDANCE ON HOW TO BUILD EVODEVO SYSTEMS

Contributions

RQ1: *"What kind of information must be present in the genome in order to produce computation in any of the computational classes?"*

- *Genome parameters: plausible indication of developmental properties*
- *Do not guarantee developmental behavior (the other way around)*
- *Parameters generalized for different dimensionalities, CA size, cell types*
- *Genome sub-transitions and sub-parameters (death – growth)*

Contributions

RQ2: *"How to quantify developmental complexity, i.e. emergent phenotypic complexity?"*

- *As measure of phenotypic & developmental properties: developing organism as a whole, phenotypic changes*
- *Trajectory and attractor length: abstract (application / computational task independent)*
- *Approx. of Kolmogorov Complexity: compression algorithms*

Contributions

RQ3: *"Do genome parameters give any information on the evolvability of the system? And if yes, can genome information be used to guide evolutionary search in favourable areas of the search space where the wanted emergent behavior is more likely to be found?"*

- *Genomes with given parameter value are likely to evolve to similar behaviors, as long as offspring has similar parameter value (evolvability)*
- *Parameters (as Lambda or sub-transitions) to guide evolution*

Contributions

RQ4: *"How can scalability of artificial EvoDevo systems be improved towards achieving systems that can fully unleish their inherent complexity, e.g. potentially at the levels of complexity found in nature?"*

- *Complexification: evolutionary growth of genomes*
- *Indirect encoding are a necessity (if target nature levels of complexity)*
- *Gene duplication is a plausible mechanism*
- *Different genotype-to-phenotype mappings*

Further Work

- Robustness of solutions (at gtype and ptype level)
- Other parameters: Sensitivity, MFP, Z, sub-transitions
- Growth in state space (true complexification)
- Self-modifying instructions

Key References

C. Langton. Computation at the Edge of Chaos: Phase Transitions and Emergent Computation. Physica D Volume 42 (1990) pp. 12-37

S. Wolfram. Universality and Complexity in Cellular Automata. Physica D Volume 10 Issue 1-2 (1984) pp. 1-35

M. Sipper, The Emergence of Cellular Computing, Computer, vol. 32, no. 7, (1999) pp. 18-26, doi:10.1109/2.774914

S. Wolfram. A New Kind of Science. Wolfram Media Inc, (2002), 1197pages – ISBN 1-57955-008-8

J. Von Neumann. Theory and Organization of complicated automata. A. W. Burks, (1949) pp. 29-87 [2, part one]. Based on transcript of lectures delivered at the University of Illinois in December 1949.

L. Altenberg. Evolving better representations through selective genome growth. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 182–187. IEEE, 1994.

M. Bidlo and J. Skarvada. Instruction-based development: From evolution to generic structures of digital circuits. *KES Journal*, 12(3):221–236, 2008.

R. Doursat, H. Sayama, and O. Michel. *Morphogenetic Engineering: Toward Programmable Complex Systems*. Springer Publishing Company, Incorporated, 2013.

D. Federici and K. Downing. Evolution and development of a multicellular organism: Scalability, resilience, and neutral complexification. *Artificial Life*, 12:2006, 2006.

C. Langton. Self-reproduction in cellular automata. *Physica D*, 10D(1-2):135–44, 1984.

J. F. Miller and W. Banzhaf. Evolving the program for a cell: from french flags to boolean circuits. In S. Kumar and P. J. Bentley, editors, *On Growth, Form and Computers*. Academic Press, October 2003.

K. O. Stanley and R. Miikkulainen. Achieving high-level functionality through evolutionary complexification. In *Proceedings of the AAAI-2003 Spring Symposium on Computational Synthesis*, Stanford, CA, 2003. AAAI Press.

K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21:63–100, 2004.

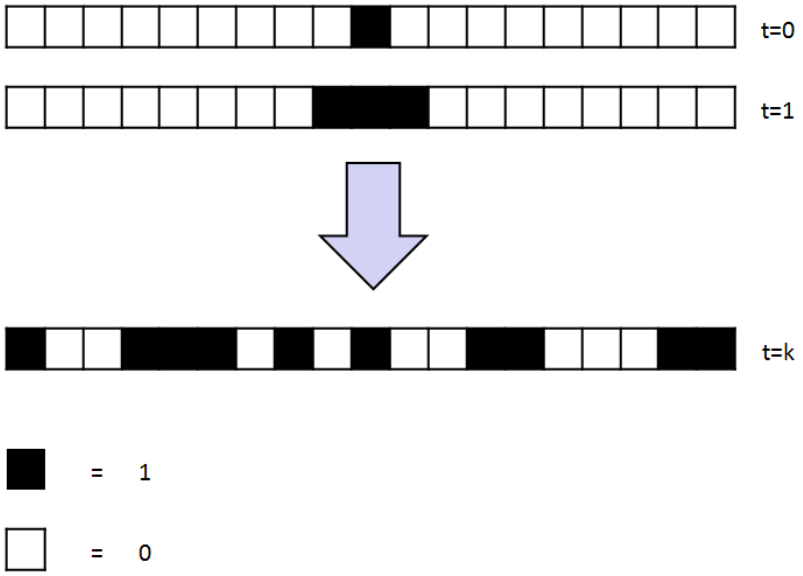
M. A. Trefzer, T. Kuyucu, J. F. Miller, and A. M. Tyrrell. On the Advantages of Variable Length GRNs for the Evolution of Multicellular Developmental Systems. *Evolutionary Computation, IEEE Transactions on*, 17(1):100–121, Feb. 2013.

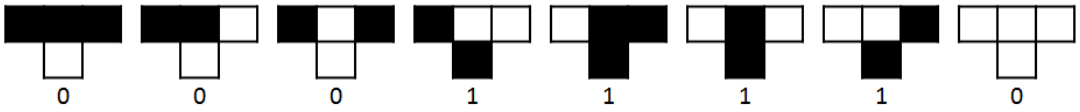
Thanks!

Questions?

Bonus Slides

Cellular Automata





Countable array of discrete cells *i*

Discrete-time update rule Φ
(operating in parallel on local neighborhoods of a given radius *r)*

Alphabet: $\sigma^i_t \in \{0, 1,..., k- 1\} \equiv A$

Update function: $\sigma^i_{t+1} = \Phi(\sigma^{i-r}_t, ..., \sigma^{i+r}_t)$

$s_t \in A^N$

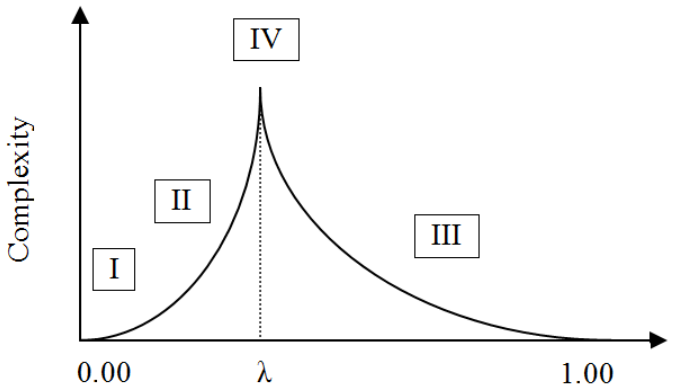
Global update Φ: $A^N \rightarrow A^N$

$s_t = \Phi\, s_{t-1}$

Lambda Parameter

$$\lambda = \frac{K^N - n}{K^N}$$

- n = number of transitions to the quiescent state (state 0)
- K = number of cells types = 3 (in our model)
- N = neighborhood size = 5 (Von Neumann neighborhood)



Majority Parameter

- how many neighborhood configurations in the rule table follow the majority state to determine the next state

$$M = \sum_{(V1V2 \ldots Vm)} [V(m+1) = maj(V1V2 \ldots Vm)]$$

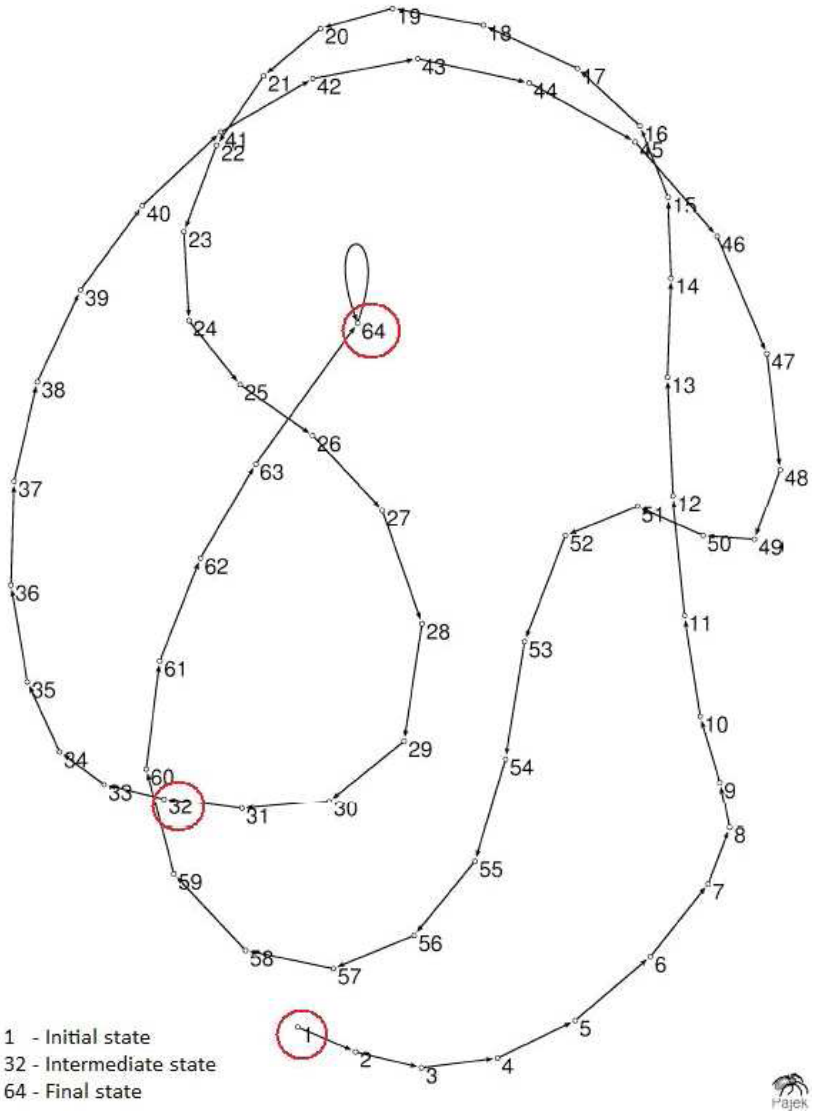
- m = number of cells in the neighborhood
- V(m+1) = value of the cell being considered, at the next time step
- maj() = function that retrieves the most present cell type (or the set of most present cell types) in the neighborhood

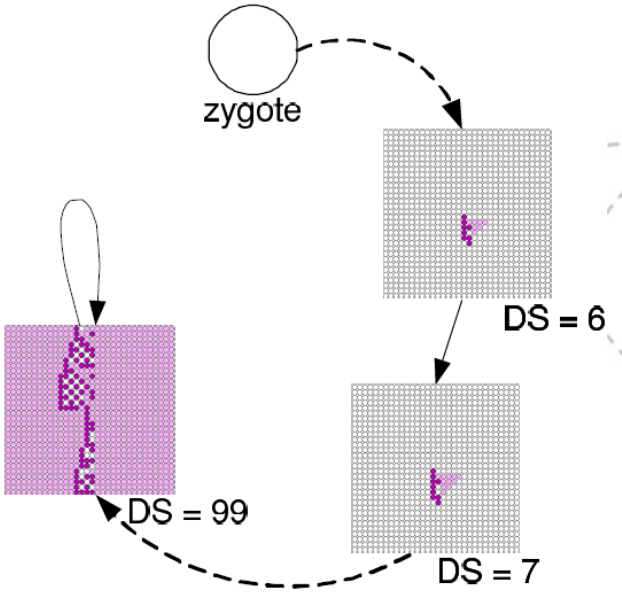
Sensitivity Parameter

- measures the number of changes in the output of the transition table based on a change in the neighborhood, one cell at a time, over all the possible neighborhoods of the rule being considered

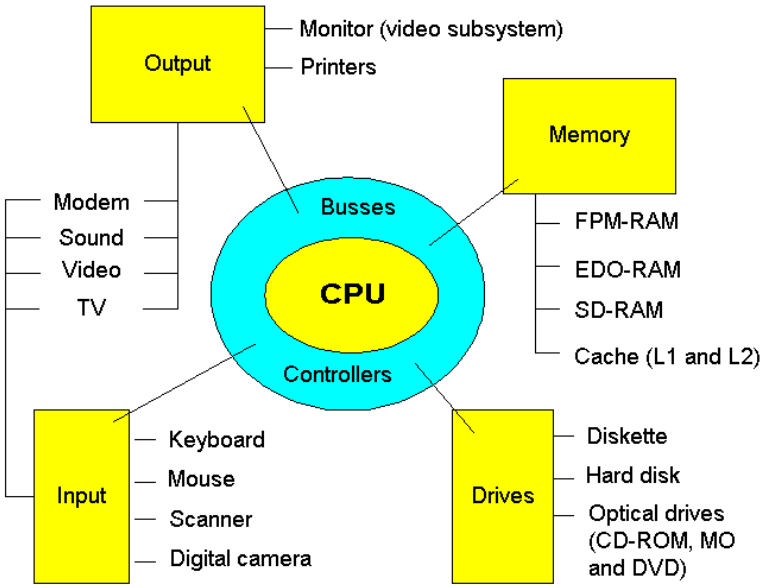
$$\mu = \frac{1}{n * m * (\mathrm{K}-1)} \sum_{(V1V2 \ldots Vm)} \sum_{q=1}^{m} \frac{\delta \varphi}{\delta Vq} \longrightarrow \frac{\delta \varphi}{\delta Vq} = \begin{cases} 1 \; if \; \varphi(V1 \ldots Vq \ldots Vm) \neq \varphi(V1 \ldots \overline{Vq} \ldots Vm) \\ 0 \; otherwise \end{cases}$$

- m = number of cells in the neighborhood
- n = possible neighborhood configurations (V1V2…Vm = 3^5 = 243)
- K = number of cell types





The developmental path shown as a trajectory.

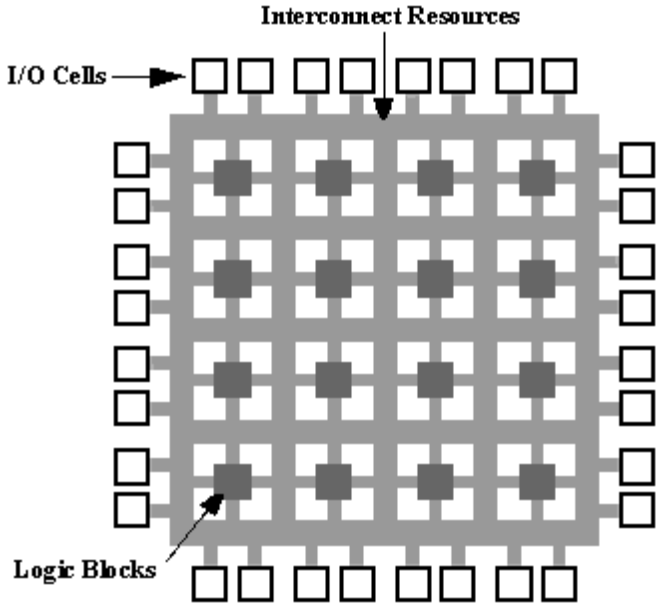


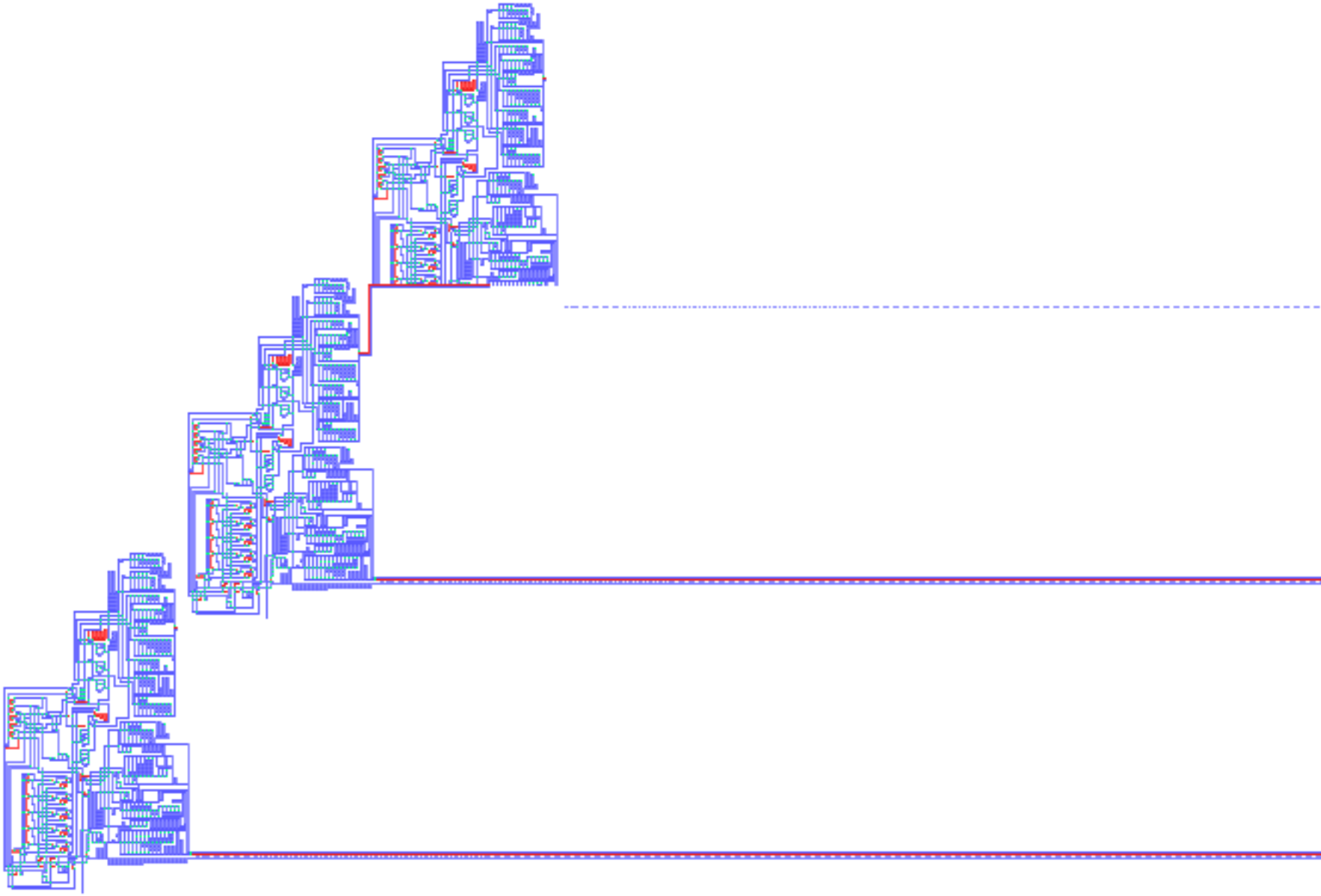
von Neumann architecture

- 1 complex processor
- tasks executed sequentially

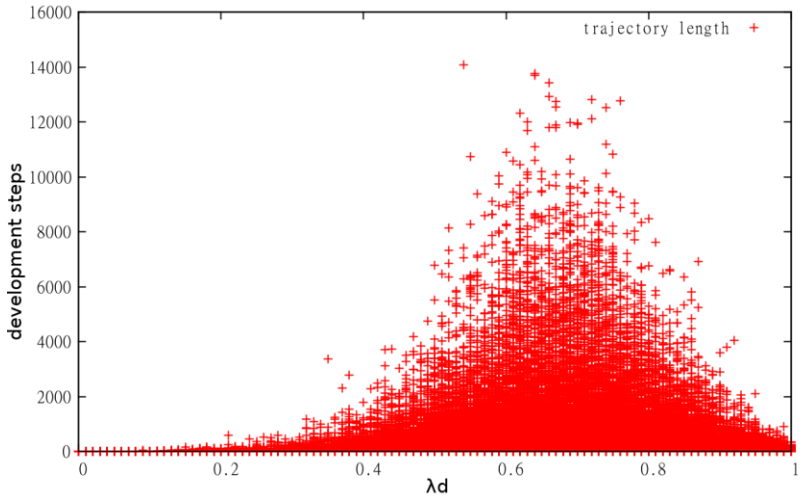
cellular computing

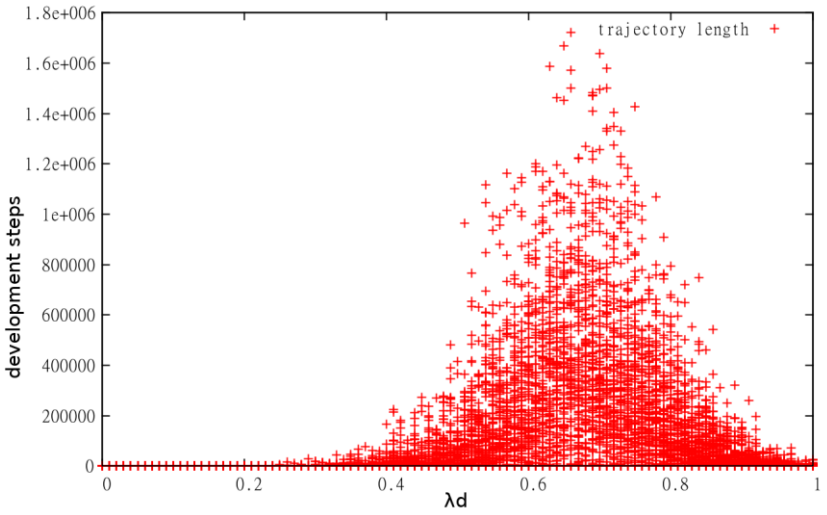
- myriad of small and unreliable parts: cells
- simple elements governed by local rules
- cells have no global view

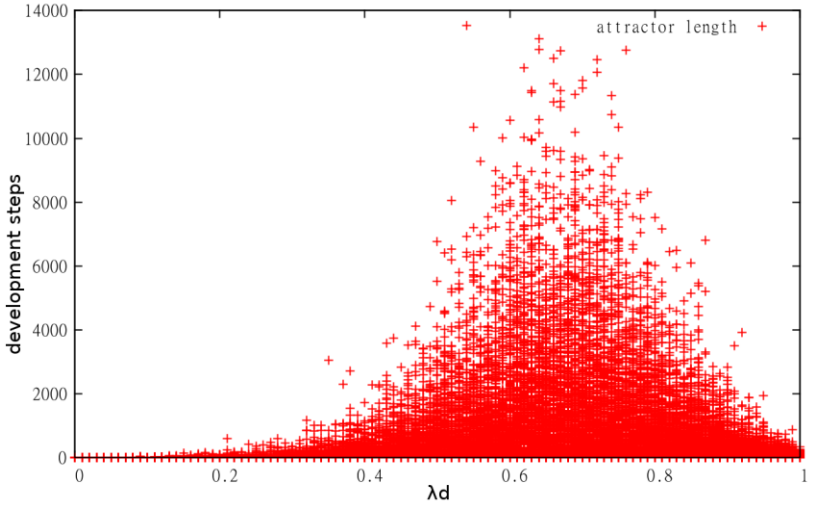


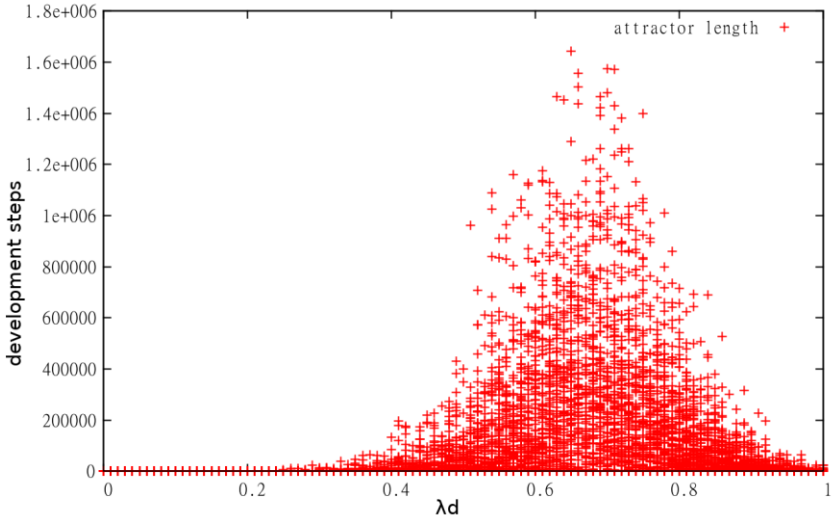


A.1





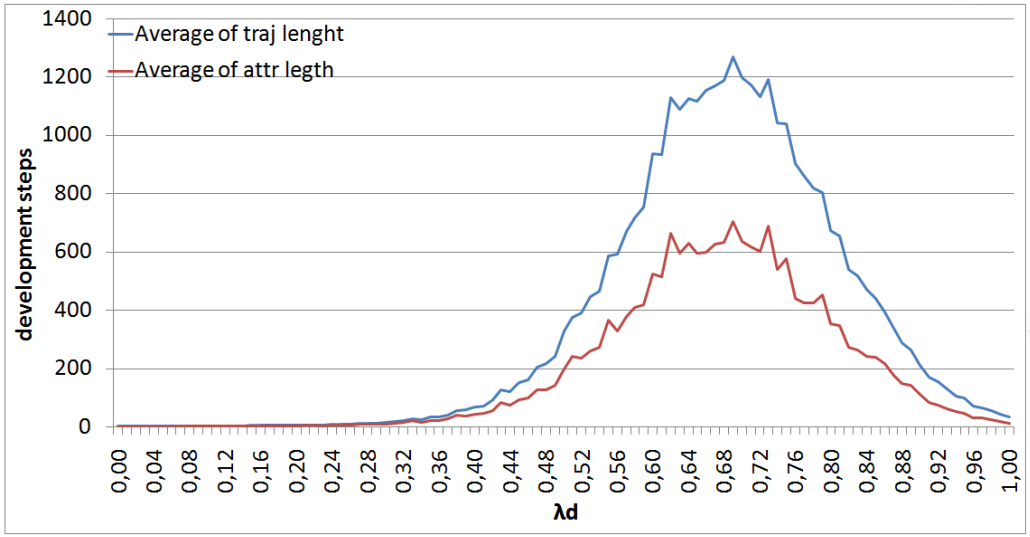




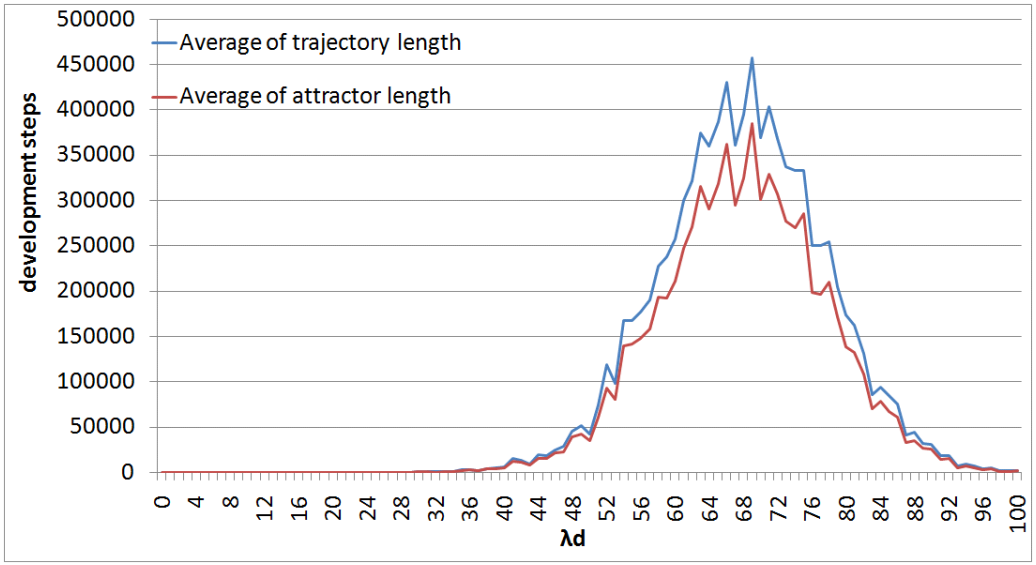
Results for 4x4 organisms plotted as function of *λd.*
1000 tests for each λd.

Results for 5x5 organisms plotted as function of *λd.*
1000 tests for each λd.

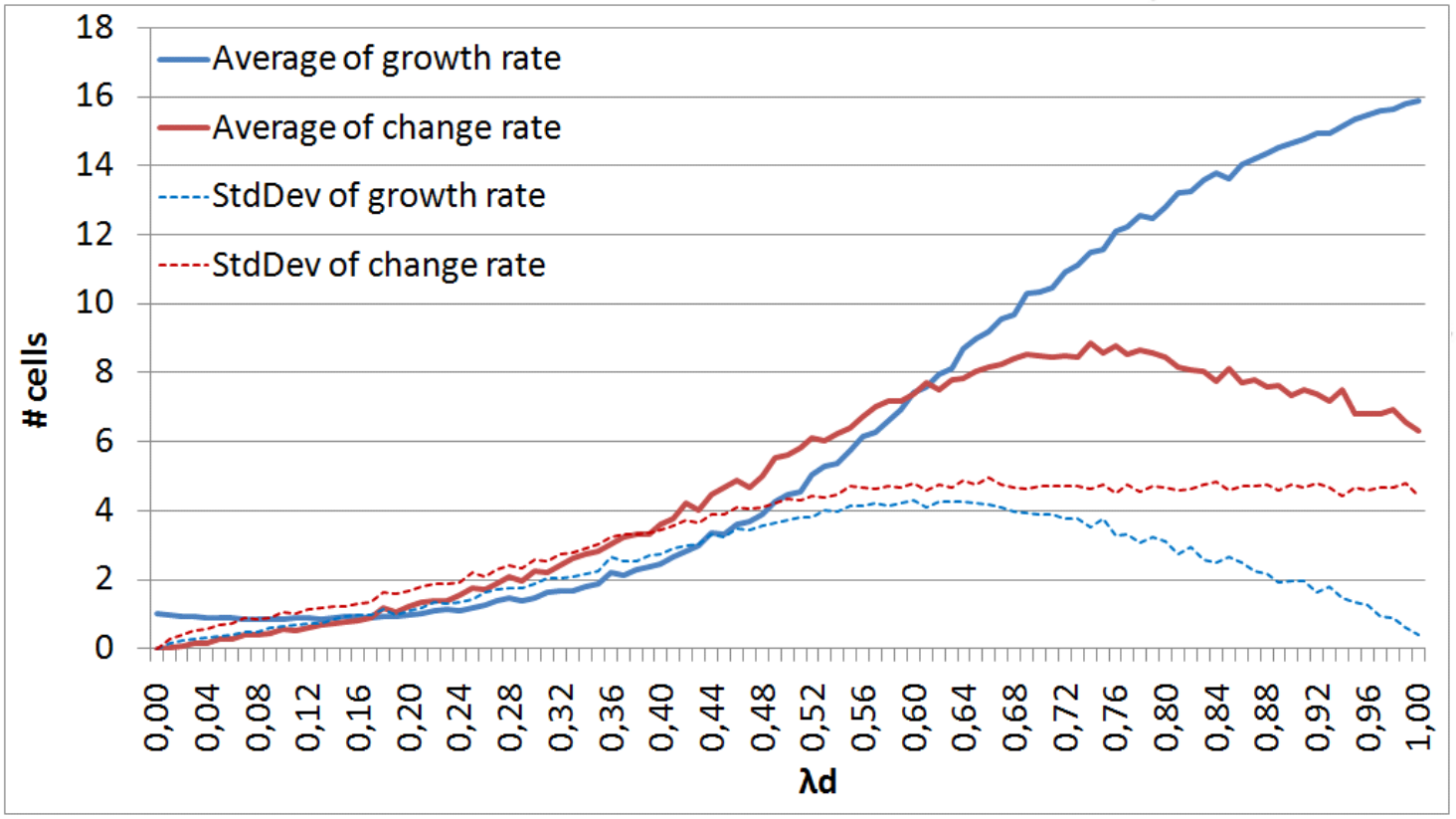
NTNU – Trondheim
Norwegian University of
Science and Technology


Average of traj lenght
Average of attr legth
development steps
1400
1200
1000
800
600
400
200
0
0,00 0,04 0,08 0,12 0,16 0,20 0,24 0,28 0,32 0,36 0,40 0,44 0,48 0,52 0,56 0,60 0,64 0,68 0,72 0,76 0,80 0,84 0,88 0,92 0,96 1,00
λd

4x4


Average of trajectory length
Average of attractor length
development steps
500000
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80 84 88 92 96 100
λd

5x5



Average growth and change rate in correlation to *λd on a 4x4 organism.*
Average over a 1000 tests for each *λd value*

B.1

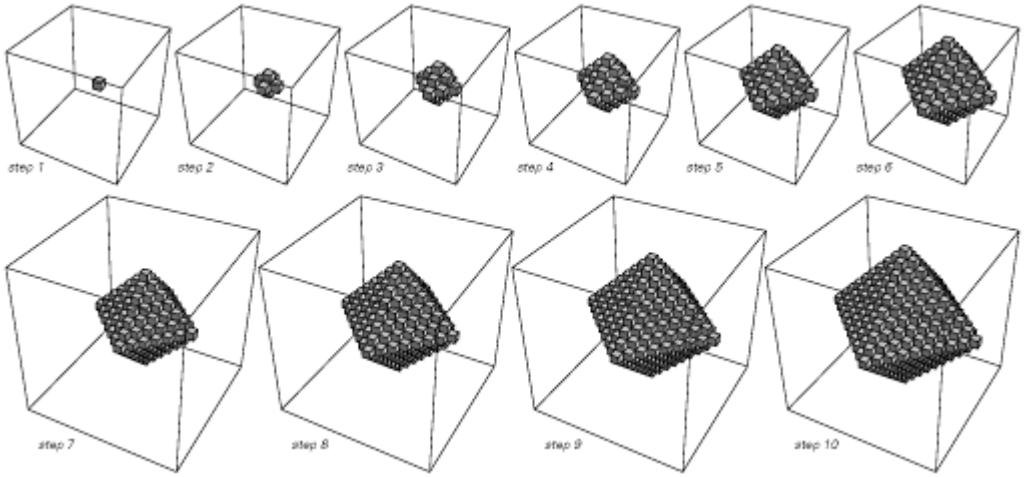


Image from ”A New Kind of Science”, Stephen Wolfram (2002), Wolfram Media

Goal

- Can genome information be used to predict emergent structural complexity?

1. Measure phenotypic structural complexity of artificial cellular organisms (approximation of Kolmogorov complexity)

2. Relate Lambda genome parameter to the measured structural complexity. Estimate developed organisms' phenotypic complexity

Kolmogorov Complexity

Image from xkcd.com

Kolmogorov Complexity

- The notion of complexity is used differently in distinct fields of computer science.
- **Definition *(Kolmogorov complexity):*** *Fix a Turing Machine U. We define the Kolmogorov function, C(x) as the length of the smallest program generating x.*

  $C(x) = min_p \{ |p| : U(p) = x\}$

- ***Invariance Theorem****: the particular choice of the universal* machine only affects *C(x) by a constant additive factor*

  $\forall x,\ C(x) \leq |x| + c$

Incomputability

- Kolmogorov complexity is incomputable. Proof by contradiction or by reduction to the non-computability of the halting problem (Turing equivalent)
- Approximations by data compression: hardly compressible strings have presumably high Kolmogorov complexity. Complexity is proportional to the compression ratio
- Incompressibility Lemma: some strings are not compressible, i.e. random strings
  Formally, a string *x is c-incompressible if* $C(x) \geq |x| - c$

Lempel-Ziv compression

- Compression algorithms tend to compress repeated patterns and structures, thus being able to detect structural features in phenotype states.
- Deflate: variation of LZ77, loseless data compression algorithm, computationally inexpensive, independent of the dimensionality of the state

- 1D CA: string representing the state of the system at a certain time step compressed directly

- 2D CA (3x3 example, same for 3D where all the rows are listed for all the depth levels)

0	1	0
1	1	2
1	0	0

$\rightarrow r = \text{“010112100”}$

$t = Deflate\ (r)$

$q = Length\ (t)$

$r_{min} = \text{”000000000”}$ $\quad q_{min} = Length(Deflate(r_{min}))$

$r_{max} = \text{”012345678”}$ $\quad q_{max} = Length(Deflate(r_{max}))$

$c = (q - q_{min}) / (q_{max} - q_{min})$

0	0	0
0	0	0
2	1	2

$\rightarrow$ T

2	0	0
1	0	0
2	0	0

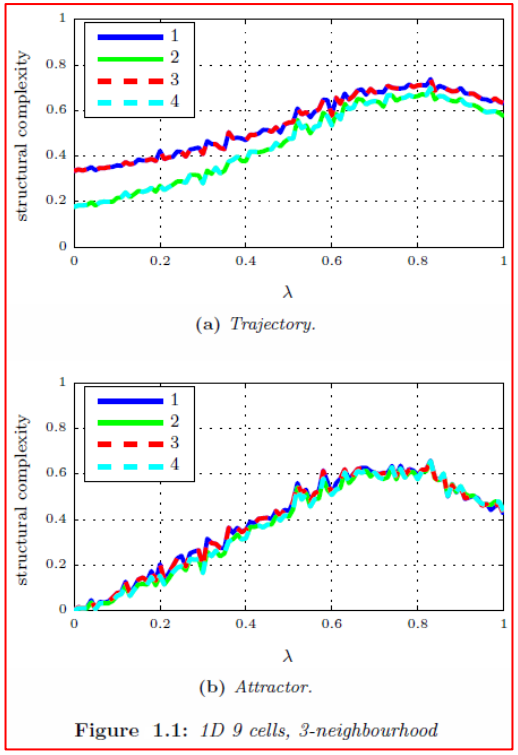
1	Simple Deflate compression	The CA state is represented as a concatenated string and directly compressed.
2	Average of all rotations	The CA state is rotated in all the possible orientations and the correspondent state strings are compressed. The average is computed.
3	Average of all translations	The CA state is shifted in all the possible positions and the correspondent state strings are compressed. The average is computed.
4	Rotations + translations	Both point 2 and 3. The CA state is rotated in all the possible orientations. Each of them is shifted in all the possible positions and the correspondent state strings are compressed. The overall average is computed.

Experimental Setup

Dimensionality	*Size*	*Cells*	*Neighborhood radius*
Experiment 1:			
1D	9	9	3
1D	9	9	5
1D	16	16	5
1D	8	8	7
2D	3x3	9	5
2D	4x4	16	5
3D	2x2x2	8	7
Experiment 2:			
1D	25	25	3
1D	27	27	3
1D	25	25	5
1D	27	27	7
2D	5x5	25	5
3D	3x3x3	27	7

State space sizes:
3by3 = 3^9 = 19.683
4by4 = 3^16 = 43.046.721
5by5 = 3^25 = 847.288.609.443

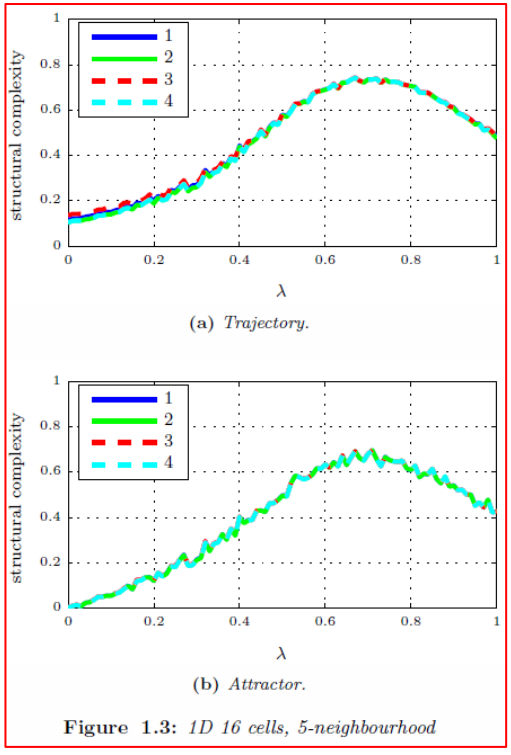
Results (exp. 1)



(a) *Trajectory.*

(b) *Attractor.*

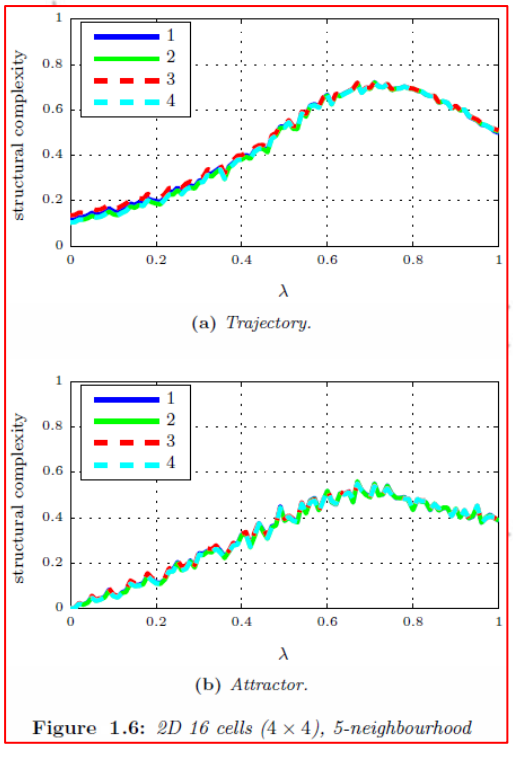
Figure 1.1: *1D 9 cells, 3-neighbourhood*



(a) *Trajectory.*

(b) *Attractor.*

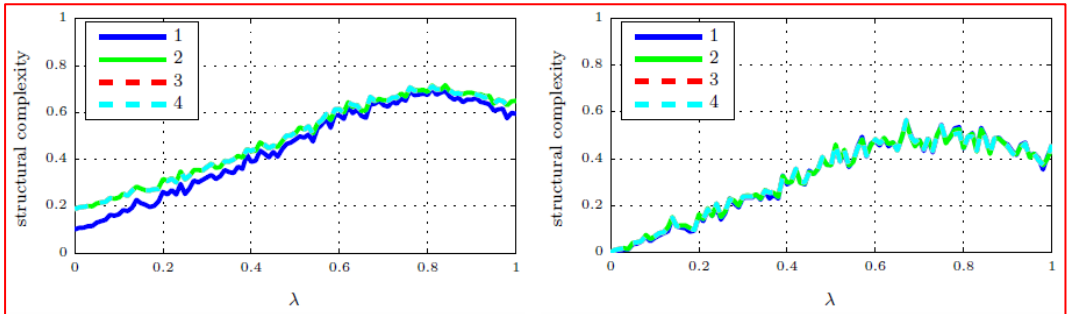
Figure 1.3: *1D 16 cells, 5-neighbourhood*



(a) *Trajectory.*

(b) *Attractor.*

Figure 1.6: *2D 16 cells (4×4), 5-neighbourhood*



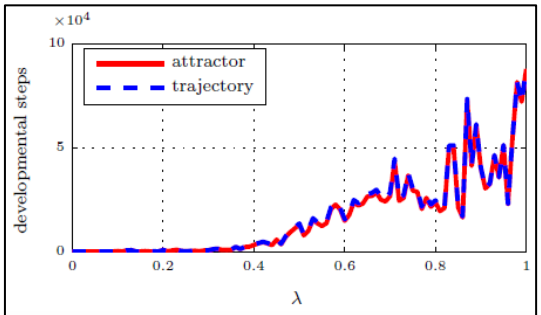
(a) *Trajectory.*

(b) *Attractor.*

Figure 1.7: *3D 8 cells ($2 \times 2 \times 2$), 7-neighbourhood*

NTNU – Trondheim
Norwegian University of
Science and Technology

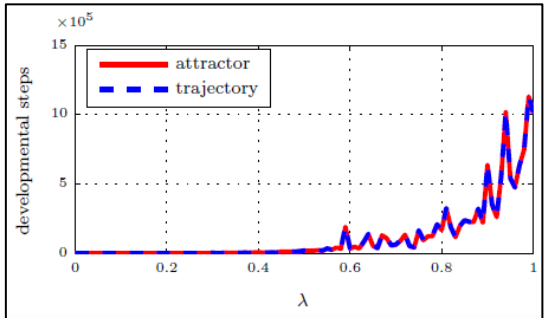
Results (exp. 2)



(a) *Average trajectory and attractor length.*

(b) *Structural complexity of trajectory and attractor.*

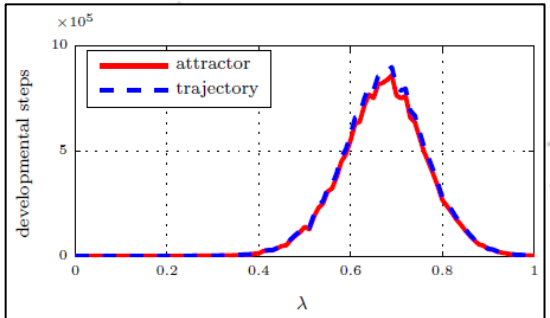
Figure 2.1: *1D 25 cells, 3-neighbourhood*



(a) *Average trajectory and attractor length.*

(b) *Structural complexity of trajectory and attractor.*

Figure 2.2: *1D 27 cells, 3-neighbourhood*

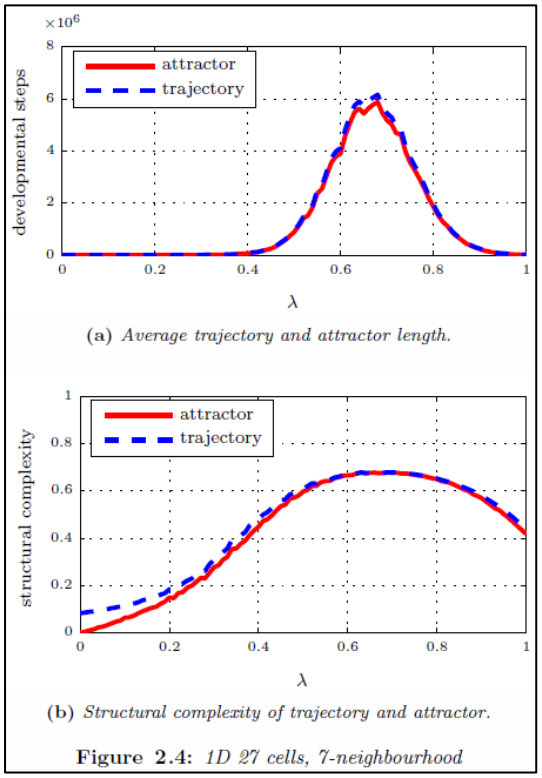


(a) *Average trajectory and attractor length.*

(b) *Structural complexity of trajectory and attractor.*

Figure 2.3: *1D 25 cells, 5-neighbourhood*

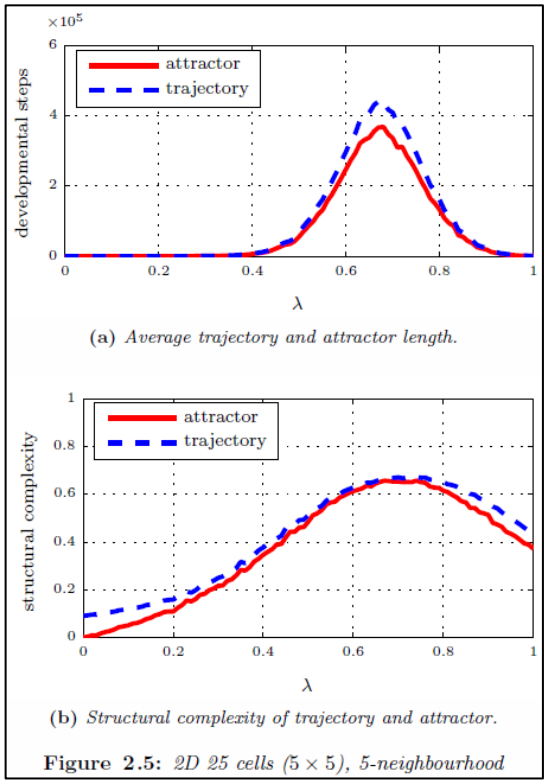
Results (exp. 2)



(a) *Average trajectory and attractor length.*

(b) *Structural complexity of trajectory and attractor.*

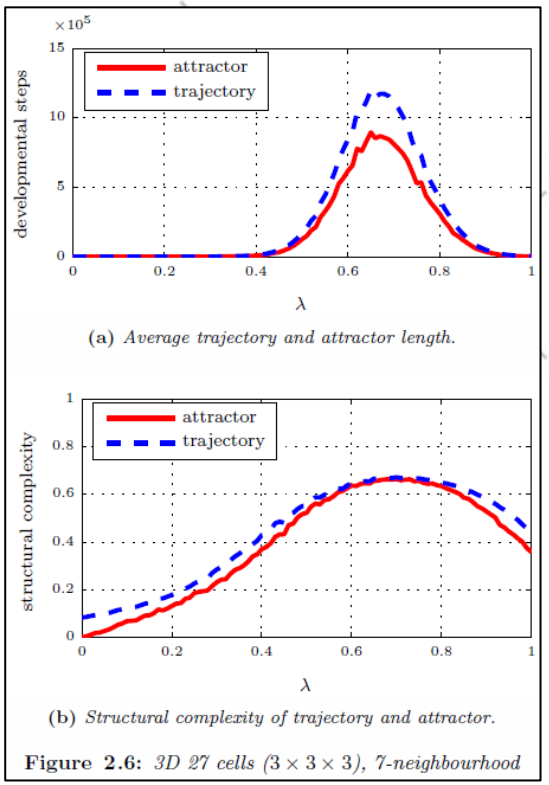
Figure 2.4: *1D 27 cells, 7-neighbourhood*



(a) *Average trajectory and attractor length.*

(b) *Structural complexity of trajectory and attractor.*

Figure 2.5: *2D 25 cells (5×5), 5-neighbourhood*



(a) *Average trajectory and attractor length.*

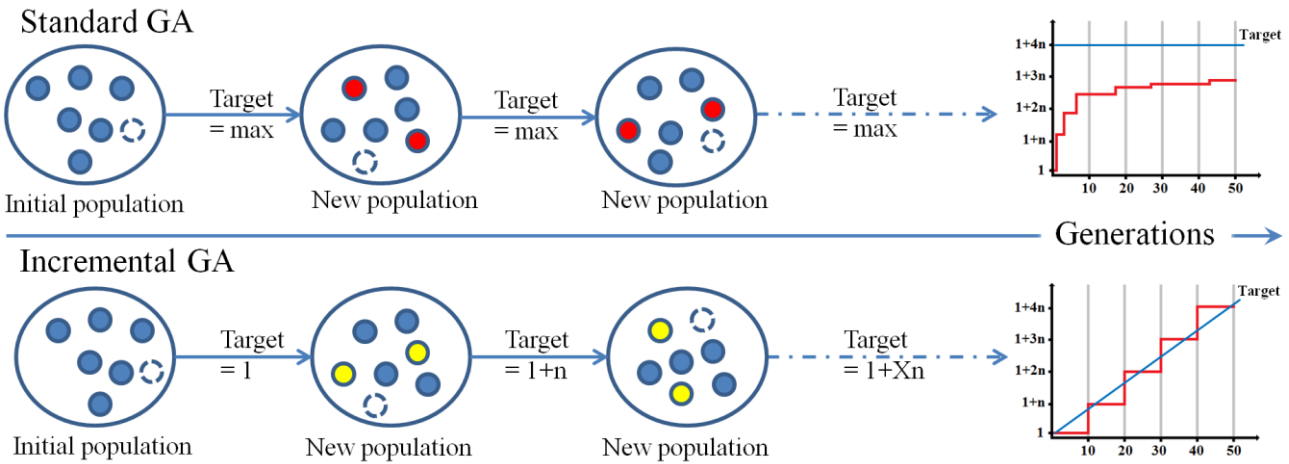
(b) *Structural complexity of trajectory and attractor.*

Figure 2.6: *3D 27 cells ($3 \times 3 \times 3$), 7-neighbourhood*

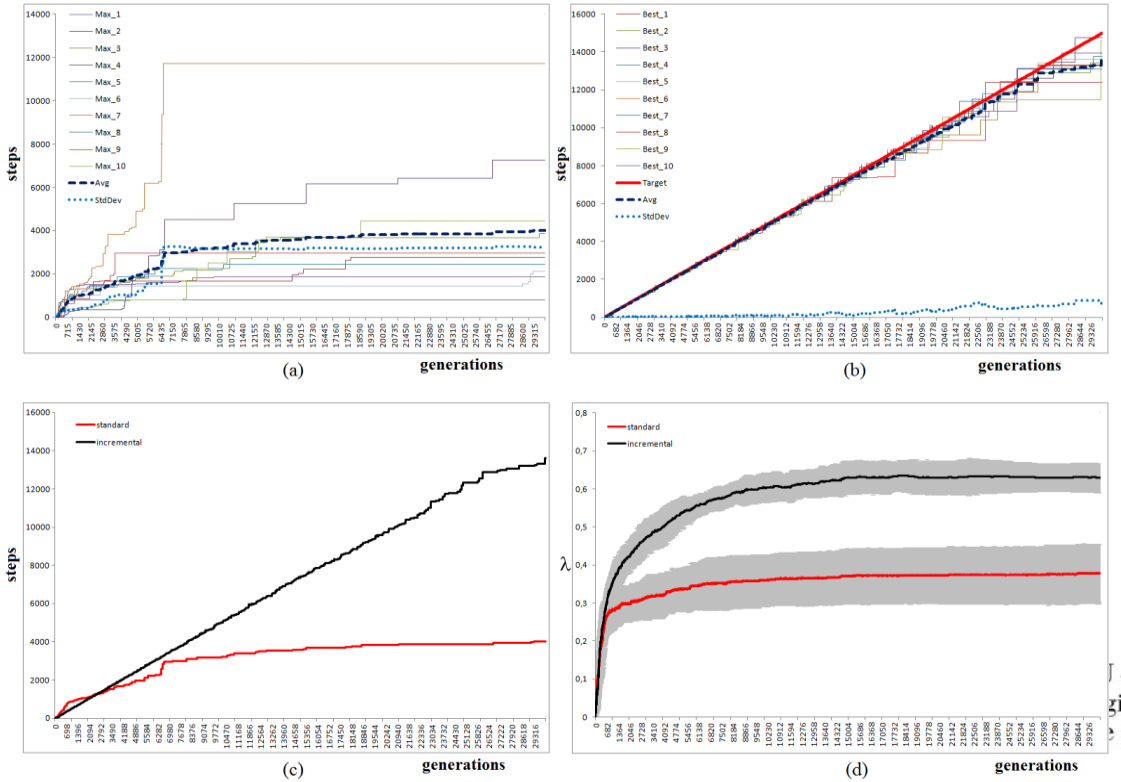
Conclusion

- Phenotypic structural complexity is strongly related to Lambda genome parameter and its ability to detect different behavioral regimes
- Dimensionality independent (1D, 2D, 3D CA)
- Possible to characterize parameter space when dimensionality, # states and neighborhood are rather small. Not possible with transient and attractor length

C.1



Results
(dead genomes)



– Trondheim
gian University of
e and Technology

Genomes generation with λ parameter A.2

Genomes generated with predefined values of λ

Similar method to Langton's random table method

For every entry in the development table:

- with probability (1- λ) the cell type at the next developmental step is quiescent (type 0)
- with probability (λ), the cell type at the next developmental step is generated by a uniform random distribution among the other cell types (type 1 or 2)

Genomes generation with *M* parameter A.2

- if there are more than 3 occurrences of a cell type:
  - with probability M the cell type at the next developmental step follows the most present cell type in the neighborhood
  - with probability 1-M the cell type at the next developmental step is generated by a uniform random distribution among the other two cell types (the minority in the neighborhood)
- If there are 2 cell types with occurrence 2
  - with probability M/2 one of those 2 cell types is chosen
  - with probability 1–M the cell type at the next developmental step has the same type as the less present cell type in the neighborhood

Genomes generation with *μ* parameter A.2

μ is easily computable for a specific development table

Much harder to generate a development table with a target μ value, because of entry dependencies

A Genetic Algorithm is used