Optimizing the creature's rule for all-to-all communication
Patrick Ediger and Rolf Hoffmann

- Modeling moving creatures in CA
  - Actions, Behavior, Cell Structure, Arbitration
- Modeling all-to-all communication
  - Communication Situations
- Genetic Procedure
- Evaluation of the evolved algorithms
- Emerged
  - Trails, Communication Spot Patterns
The task: all-to-all communication

- **General:** Find behaviour of moving creatures for a CA system which can solve a given problem

- **CA system**
  - \( n \times m \) field, \( k \) creatures
  - set of initial configurations
    - 50% with border + 50% without border (wrap-around), no obstacles
    - arbitrary start position and start direction

- **Global Task**
  - The creatures shall **exchange (all-to-all) their information in shortest time on average using certain communication situations**

- **Goal**
  - find the **best uniform behavior (algorithm how to move around) in order to fulfill the global task**
Creature: Actions

(A) if (obstacle or creature)

(B) if (conflict)

(C) (move) if not(A or B)

move and turn (Lm/Rm)

delete

copy+turn

turn (L/R)

move and turn (Lm/Rm)
Behavior of a creature

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>s',y</td>
<td>2,R</td>
<td>5,L</td>
</tr>
<tr>
<td>i</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>j</td>
<td>0,1</td>
<td>2,3</td>
</tr>
</tbody>
</table>

**State table**, defining the behavior (algorithm) of a creature

- **State graph**
- **Input state**
- **Next state**
- **Output**
Cell structure

\[ s \in \{0, 1, 2, .. n-1\} \]

- **control machine**
  - control state \( s \)
  - RuleY activator
  - RuleS next state

- **action machine**
  - direction \( d \in \{0, 1, 2, 3\} \)
  - neighbors
  - move? (delete/copy)

- **input machine**
  - type \( \in \{\text{EMPTY}, \text{OBSTACLE}, \text{CREATURE}\} \)
  - \( v \pm 1 \)

- **RuleS**
  - next state
  - activator \( v \)

- **RuleY**
  - change

\[ \epsilon \{\text{toNorth, toEast, toSouth, toWest}\} = \{0, 1, 2, 3\} \]
Conflictdetection/arbitration by **EMPTY** cell

```
d = direction

CREATURE wants to move to the right

test whether there is exactly one creature which wants to move to "ME" \( \rightarrow \) move by copy
```
Conflict detection/arbitration by **CREATURE**

**CREATURE** wants to move to the right

Test whether there is no other creature which wants to move to the same **EMPTY** cell in front → move by delete
Arbitration logic of the EMPTY cell can be used by its neighbors.

Advantage: the arbitration logic is not replicated → 1/5 resources.

This logic is not necessary because the same logic residing in the front cell can be used.
Generalization: Extension of the CA model

- Modification of the CA model:
  - a cell can compute decision functions which can be used by the neighbors.
- Thereby the neighborhood is indirectly extended

read access

4 actual neighbors

1 actual neighbor (which evaluates 4 neighbors)
Modeling: All-to-all communication

- Information of creature $i$ represented by an additional bitvector
- Initially bit $i$ set to one
- Information exchange between creatures by ORing the bitvectors

| creature 1  | 0100000000000000 |
| creature 15 | 0000000000000001 |
| OR          | 0100000000000001 |

1111111111111111
Communication situations

(a) (b) (c) allowed

(d) (e) (f) not allowed

C = communicator

(average, priority, maximum, OR)
Genetic procedure (1)

- *P populations of N individuals* are updated in each iteration (cycle)
- During each cycle *M offsprings* are produced in each population.
- The *union* of the current *N* individuals and the *M* offsprings
  - sorted according to their fitness
  - *N* best are selected forming the next population.

- **Fitness**: Evaluation by simulation
  1. *success rate*
     - percentage of successfully (all-to-all comm.) solved configurations
  2. *speed* (no. of gen,)
     - All-to-all communication as fast as possible
Genetic procedure (2): Parents

(1. Get Parents)

- Two parents are chosen for each population.
- Each parent is chosen from the own population with a probability of $p_1$ and from an arbitrary other population with the probability of $(1 - p_1)$.
Genetic procedure (3): Offsprings

- (2. Crossover)
  - Each new component \((si, yi)\) of the genome string is taken from either the first parent or the second parent with a probability of 50%.

\[
(i, y) = \text{(nextstate, output)}
\]
Genetic procedure (4): Offsprings

- **(3. Mutation)**
  - The string being modified by the crossover is afterwards mutated with a probability of $p_2$.
  - If a mutation shall be performed, an *arbitrary position* $j$ is chosen and a new value (randomly chosen from the set of valid values) is replacing the existing one.
  - Thereby either the next state or the output is changed at position $i$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$s',y$</td>
<td>1,R</td>
<td>5,L</td>
<td>3,L</td>
<td>4,R</td>
<td>5,R</td>
<td>3,L</td>
</tr>
<tr>
<td>$i$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$j$</td>
<td>0,1</td>
<td>2,3</td>
<td>4,5</td>
<td>6,7</td>
<td>8,9</td>
<td>10,11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$1$</td>
<td>1,Lm</td>
<td>2,Rm</td>
<td>3,Rm</td>
<td>4,Lm</td>
<td>5,Rm</td>
<td>0,Lm</td>
</tr>
<tr>
<td>$i$</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>$j$</td>
<td>12,13</td>
<td>14,15</td>
<td>16,17</td>
<td>18,19</td>
<td>20,21</td>
<td>22,23</td>
</tr>
</tbody>
</table>
Parameter settings

P = 7 populations with N = 100 individuals each
M = 10 offsprings
p1 = 98% (from own population)
1-p1 = 2% (from other population)
p2 = 9% (mutation)

k = 16 creatures,
n x m = 33 x 33 = 1089 cells
50% initial configurations with border, 50% with wrap-around
no obstacles
Initial configurations used

<table>
<thead>
<tr>
<th>Initial configs</th>
<th>#</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training set</strong></td>
<td>10</td>
<td>after 6000 iterations $\rightarrow$ 700 algorithms evolved ($ALGS$)</td>
</tr>
<tr>
<td><strong>Ranking set</strong></td>
<td>100</td>
<td>ranking ($ALGS$) $\rightarrow$ $A1, B1, C1, ...$</td>
</tr>
<tr>
<td><strong>Robustness set</strong></td>
<td>80</td>
<td>additional fitness check ranking ($ALGS$) $\rightarrow$ $A2, B2, C2, ...$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>different grid sizes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1089 x 1, 363 x 3, 121 x 9, 99 x 11</td>
</tr>
</tbody>
</table>
Best evolved algorithm \( A_1 = A_2 \)

\[ 2R5L3L5L2R5R-3L4L0L1R5R2R \]

<table>
<thead>
<tr>
<th></th>
<th>success rate ( 100% )</th>
<th>generations (mean)</th>
<th>comm. spots a+b+c</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking Set</td>
<td>100/100</td>
<td>773.21 (faster)</td>
<td>209.9 (more effective)</td>
</tr>
<tr>
<td>Robustness Set</td>
<td>80/80</td>
<td>1 905.61</td>
<td>378.26</td>
</tr>
</tbody>
</table>

**communication spot**: a site which is read in a communication situation.
Actions if creature cannot move
Actions if creature can move

(LmRmLmRmRmLm)*
How robust are the algorithms?

- All 100 algorithms are successful for 37 configurations.
- Only 21 algorithms are successful for 61 configurations.
- Only 1 algorithm is successful for 80 configurations.

The graph shows the number of algorithms that are successful for different numbers of successfully solved initial configurations. The more difficult environments of the Robustness set (1089 x 1) are highlighted.
Communication with the Evolved Alg. is much better than with Random Walk

![Graph showing comparison between Evolved Alg. A1=A2 and Random Walk.](image)
Emerged trails, Alg. A1=A2

border

wrap

border

wrap

0  135  275  483

0  200  400  898
Centers of information

Alg. B2, wrap-around, generation 400

creatures are circulating in little squares → centers of information
Evolved patterns for communication (communication spots)

Alg. B2, wrap-around, last generation 898

in parallel to the "axes"

*communication spot.*

a site which is read in a communication situation
Conclusion

- CA rule for moving creatures
  - Behavior modeled by an internal hidden automaton

- Conflict detection logic needs not to be replicated
  - generalized: extension of the CA model

- Goal
  - How to move around in order to communicate all-to-all

- Evolved algorithms
  - only one alg. found which can solve all given configurations (Training + Ranking + Robustness set)
  - Evolved creatures are much better than Random Walkers
  - "wrap-around" is easier for random walkers – "with border" is easier for evolved creatures

- Emerged
  - Trails: Weaving patterns with centers of information
  - Communication spot patterns: in parallel to the borders
Future work

- more powerful actions
- more control states
- time-shuffled algorithms
- non-uniform creatures
- more complex communication mechanisms
- complete quality test / formal proves for very small configurations
Add On: Control machine and action machine

move signal from front cell

\[ x = m \]

**MEALY-Control-Machine**

- States:
  - 0: L
  - 1: L
  - 2: L
  - 3: R
  - 4: R
  - 5: R
  - 6: Lm
  - 7: Lm
  - 8: Rm
  - 9: Rm

**Rule S**

if \( y = 1 \) then \( d := d + 1 \) (R)
else \( d := d - 1 \) (L)

**Rule Y = y**

**Copy states to front cell if \( m = \text{true} \)**

**MOORE-Action-Machine**

- States:
  - s: control state
  - d: direction
  - v(d,y): new direction
  - v: creature can move
  - L/R: turn left/R if \( (m=1) \)
  - Lm/Rm: turn left/R and move if \( (m=0) \)

**Move direction to neighbors**
Add On: Results (separately evolved border/wrap)

- **with border**
  - diagonal trails
  - fast

- **wrap-around**
  - mesh, turned
  - slow