How efficient are creatures with time-shuffled behaviors?

Patrick Ediger, Rolf Hoffmann and Mathias Halbach

- Creatures' Exploration Problem
- CA Model
  - Rule
  - Cell Type
  - Actions of the Creature
  - Conflicts, 1-Phase Arbitration
  - Different Behaviors of a Creature
- Success rates of the best algorithms for one creature
- Time-shuffled behavior
  - Shuffle Modes
  - Success Rates for one Creature
- Multi creature systems
  - Success Rates
  - Efficiency Measures
  - Other Aspects of mixing behaviors
- Conclusion
Creatures' Exploration Problem

The Problem
- Given is a 2D-CA with obstacles and moving creatures.
- Goal: Find an optimal local rule for the creatures to visit a maximum number of empty cells with a minimum number of time steps for a given set of initial configurations.

Applications
- Mowing a lawn in shortest time
- Vacuum cleaning a room
- Exploring an unknown environment
- Distribute „information“
# The Actions of the Creatures

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L</strong></td>
<td>(turn Left)</td>
</tr>
<tr>
<td><strong>R</strong></td>
<td>(turn Right)</td>
</tr>
<tr>
<td><strong>Lm</strong></td>
<td>(turn Left and move)</td>
</tr>
<tr>
<td></td>
<td>move forward and simultaneously turn left</td>
</tr>
<tr>
<td><strong>Rm</strong></td>
<td>(turn Right and move)</td>
</tr>
<tr>
<td></td>
<td>move forward and simultaneously turn right</td>
</tr>
</tbody>
</table>

---

![Diagram of actions](image)
The Rule for the Creatures

Legend

- obstacle or creature
- irrelevant
- creature in one out of two directions

FrontCell

(a1) if (obstacle or creature) then turn (L/R)

(a2) if (collision) then turn (L/R)

(b) if not((a1) or (a2)) then turn and move (Lm/Rm)
Cell Type

- **Type**
  - **EMPTY**
  - **OBSTACLE**
  - **CREATURE**
    - **Direction r**
      - (toNorth, toEast, toSouth, toWest)
    - **State Table**
    - **Control State s**
Conflict Resolution

- More than one creature wants to move to the same cell
- Result of conflict resolution in general
  - either **no creature** can move, or
  - **one creature** is selected to move
- Implementations
  - 2-phase arbitration
    1. check for conflict and select one creature
    2. the selected creature will move
  - 1-phase arbitration (our solution, see next slide)
1-Phase Arbitration

- **front cell** contains a **feed-back logic** for the arbitration
- evaluates the number $Q$ of requests
  - if $(Q>1)$ then $grant = 0$ else $grant = 1$
- **grant** is computed during the current clock cycle
Different Behaviors of a Creature

- Modeled by a changeable **Control Machine** and a fixed **Action Machine**
- Control Machine implements **N-State Algorithm**

**grant signal** from neighbor in front (front cell)

\[ m \]

<table>
<thead>
<tr>
<th>m</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>Lm</td>
<td>Lm</td>
<td>Lm</td>
<td>Lm</td>
</tr>
<tr>
<td>1</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>Lm</td>
<td>Lm</td>
<td>Lm</td>
<td>Lm</td>
</tr>
<tr>
<td>2</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>Lm</td>
<td>Lm</td>
<td>Lm</td>
<td>Lm</td>
</tr>
<tr>
<td>3</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>Lm</td>
<td>Lm</td>
<td>Lm</td>
<td>Lm</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>Rm</td>
<td>Rm</td>
<td>Rm</td>
<td>Rm</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>Rm</td>
<td>Rm</td>
<td>Rm</td>
<td>Rm</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>Rm</td>
<td>Rm</td>
<td>Rm</td>
<td>Rm</td>
</tr>
</tbody>
</table>

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

\[ s \rightarrow r \rightarrow v \rightarrow s' \]

**MEALY-Control-Machine**

**MOORE-Action-Machine**

if \( d = 1 \) then
- \( r := r + 1 \) (turn right)
else
- \( r := r - 1 \) (turn left)
Representation of an N-State Algorithm

- state/output-table ↔ state/output-graph

- string representation

\[1L2L0L4R5R3R-3Lm1Rm5Lm0Rm4Lm2Rm = 1L2L0L4R5R3R-3L1R5L0R4L2R\] simplified

\[
\text{# Algorithms } = (\#s \#y) (\#s \#x)
\]
Environment Test Set

- 16 environments
  - 33x33 cells with 129 obstacles and $R = 960$ empty cells
  - **Manually** designed symmetrical:
  - **Manually** designed asymmetrical:
  - **Randomly** generated:
Success rates of the best 6-state algorithms for one creature

- **Success rate** := number of successful (100%) visited environments
- used: 10 best algorithms A to J from former investigation

![Table showing success rates of algorithms](image)

- Mean over all environments: 65.63%

**Legend**

- `X` successful (100 % visited)
- `O` unsuccessful

**Table headers**

- Algorithm
- Success Env0
- Success Env1
- Success Env2
- Success Env3
- Success Env4
- Success Env5
- Success Env6
- Success Env7
- Success Env8
- Success Env9
- Success Env10
- Success Env11
- Success Env12
- Success Env13
- Success Env14
- Success Env15
- No. Successful
- Total visiting percentage

**Table content**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Success Env0</th>
<th>Success Env1</th>
<th>Success Env2</th>
<th>Success Env3</th>
<th>Success Env4</th>
<th>Success Env5</th>
<th>Success Env6</th>
<th>Success Env7</th>
<th>Success Env8</th>
<th>Success Env9</th>
<th>Success Env10</th>
<th>Success Env11</th>
<th>Success Env12</th>
<th>Success Env13</th>
<th>Success Env14</th>
<th>Success Env15</th>
<th>No. Successful</th>
<th>Total visiting percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>7</td>
<td>88.74%</td>
</tr>
<tr>
<td>G</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>7</td>
<td>87.32%</td>
</tr>
<tr>
<td>J</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>7</td>
<td>85.89%</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>6</td>
<td>86.92%</td>
</tr>
<tr>
<td>E</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>4</td>
<td>83.42%</td>
</tr>
<tr>
<td>A</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>3</td>
<td>82.38%</td>
</tr>
<tr>
<td>F</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>2</td>
<td>73.08%</td>
</tr>
<tr>
<td>D</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>0</td>
<td>47.55%</td>
</tr>
<tr>
<td>H</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>0</td>
<td>36.20%</td>
</tr>
<tr>
<td>I</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>0</td>
<td>36.20%</td>
</tr>
</tbody>
</table>
Time-shuffled behavior

- 3 different shuffling modes (c, s and a)
  - Toggling algorithms X and Y generation-wise (t = 0/1)

- **common (c)**
  - Only one common state
  - State transition and output are toggled

- **simultaneous (s)**
  - Two states always updated
  - Output is toggled

- **alternate (a)**
  - Two states
  - State transition and output are toggled
Success rates of time-shuffled behaviors for one creature

### Uniform (u) = not time-shuffled

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>No. Successful</th>
<th>Total visiting percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>7</td>
<td>88.74%</td>
</tr>
<tr>
<td>G</td>
<td>7</td>
<td>87.32%</td>
</tr>
<tr>
<td>J</td>
<td>7</td>
<td>85.89%</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>86.92%</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>83.42%</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>82.38%</td>
</tr>
<tr>
<td>F</td>
<td>2</td>
<td>73.08%</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>47.55%</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>36.20%</td>
</tr>
<tr>
<td>I</td>
<td>0</td>
<td>36.20%</td>
</tr>
</tbody>
</table>

### Time-shuffled (Top 3 of each of the shuffling modes)

<table>
<thead>
<tr>
<th>Algorithm X</th>
<th>Algorithm Y</th>
<th>Success Env0</th>
<th>Success Env1</th>
<th>Success Env2</th>
<th>Success Env3</th>
<th>Success Env4</th>
<th>Success Env5</th>
<th>Success Env6</th>
<th>Success Env7</th>
<th>Success Env8</th>
<th>Success Env9</th>
<th>Success Env10</th>
<th>Success Env11</th>
<th>Success Env12</th>
<th>Success Env13</th>
<th>Success Env14</th>
<th>Success Env15</th>
<th>No. Successful</th>
<th>Total visiting percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>J</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>12</td>
<td>85.84%</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>G</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>11</td>
</tr>
<tr>
<td>J</td>
<td>B</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>J</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>F</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>8</td>
</tr>
<tr>
<td>E</td>
<td>C</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>B</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>8</td>
</tr>
</tbody>
</table>

- higher success rate with all time-shuffle modes compared to uniform
  1. alternate
  2. simultaneous
  3. common
  4. uniform

more successful
Evaluation for several creatures

- Creatures: $k = 1 \ldots 64$, equally distributed along the borders (powers of 2, plus 12, 28 and 60)
- All pairs of algorithms A-J used with all shuffling modes
- Notation of systems:
  - Uniform: $X-k$
    - $X$: algorithm
    - $k$: number of creatures
  - Time-shuffled: $XYp-k$
    - $X/Y$: algorithms
    - $p$: shuffling mode (c, s, a)
    - $k$: number of creatures
Success rates for several creatures

- **Completely successful** (successful on all 16 environments)

<table>
<thead>
<tr>
<th>k</th>
<th>uniform</th>
<th>common</th>
<th>simultaneous</th>
<th>alternate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2,4</td>
<td>-</td>
<td>0/10</td>
<td>0/90</td>
<td>0/100</td>
</tr>
<tr>
<td>8</td>
<td>J</td>
<td>1/10</td>
<td>1/90</td>
<td>3/90</td>
</tr>
<tr>
<td>12</td>
<td>B,C,J</td>
<td>3/10</td>
<td>1/90</td>
<td>5/90</td>
</tr>
<tr>
<td>16</td>
<td>B,C,E,G,J</td>
<td>5/10</td>
<td>2/90</td>
<td>7/90</td>
</tr>
<tr>
<td>28</td>
<td>B,C,D,E,I,J</td>
<td>6/10</td>
<td>6/90</td>
<td>8/90</td>
</tr>
<tr>
<td>32</td>
<td>C,D,E,F,G,H,I,J</td>
<td>8/10</td>
<td>7/90</td>
<td>6/90</td>
</tr>
<tr>
<td>60</td>
<td>B,C,D,E,F,G,H,I,J</td>
<td>9/10</td>
<td>12/90</td>
<td>14/90</td>
</tr>
<tr>
<td>64</td>
<td>B,C,D,E,F,G,H,I,J</td>
<td>9/10</td>
<td>13/90</td>
<td>20/90</td>
</tr>
</tbody>
</table>
New metrics for several creatures

- Only **completely successful** systems will be compared.

- **step rate** := the number of generations $g_{\text{max}}$ needed to visit one new empty cell with $k$ creatures, averaged over environments.

  \[
  \text{step rate} := \frac{g_{\text{max}}}{R}
  \]
  Sum of all empty cells of all environments

- **normalized work** := average steps needed by all $k$ creatures to visit one new empty cell (cost for each visited cell).

  \[
  \text{normalized work} = \text{nw} := k \times \text{step rate}
  \]

- **absolute efficiency** := the cost compared to a reference system $XYp-k_{\text{min}}$.

  \[
  \text{absolute efficiency} := \frac{\text{nw}_{XYp-k_{\text{min}}}}{\text{nw}}
  \]

  1. find completely succ. system with $k_{\text{min}}$
  2. use most efficient uniform system
The fastest systems

- Increase of speed by using time-shuffling? (→ Yes, but limited)

Most of the fastest systems are uniform (J, B, C, G)

<table>
<thead>
<tr>
<th>No. of Creatures (k)</th>
<th>Algorithm X</th>
<th>Algorithm Y</th>
<th>Shuffle Mode</th>
<th>mean g_max</th>
<th>mean normalized work</th>
<th>absolute efficiency compared to J-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>B</td>
<td>C</td>
<td>c</td>
<td>218</td>
<td>14,51 €</td>
<td>1,038</td>
</tr>
<tr>
<td>60</td>
<td>J</td>
<td>J</td>
<td>u</td>
<td>224</td>
<td>14,03 €</td>
<td>1,073</td>
</tr>
<tr>
<td>60</td>
<td>B</td>
<td>C</td>
<td>c</td>
<td>233</td>
<td>14,58 €</td>
<td>1,033</td>
</tr>
<tr>
<td>60</td>
<td>G</td>
<td>G</td>
<td>u</td>
<td>245</td>
<td>15,34 €</td>
<td>0,982</td>
</tr>
<tr>
<td>64</td>
<td>J</td>
<td>J</td>
<td>u</td>
<td>257</td>
<td>17,15 €</td>
<td>0,878</td>
</tr>
<tr>
<td>60</td>
<td>C</td>
<td>C</td>
<td>u</td>
<td>257</td>
<td>16,09 €</td>
<td>0,936</td>
</tr>
<tr>
<td>64</td>
<td>B</td>
<td>B</td>
<td>u</td>
<td>261</td>
<td>17,38 €</td>
<td>0,867</td>
</tr>
<tr>
<td>60</td>
<td>B</td>
<td>B</td>
<td>u</td>
<td>276</td>
<td>17,25 €</td>
<td>0,873</td>
</tr>
<tr>
<td>64</td>
<td>C</td>
<td>C</td>
<td>u</td>
<td>285</td>
<td>18,99 €</td>
<td>0,793</td>
</tr>
<tr>
<td>64</td>
<td>G</td>
<td>G</td>
<td>u</td>
<td>314</td>
<td>20,92 €</td>
<td>0,720</td>
</tr>
</tbody>
</table>

mean over all environments

- But: Fast does not necessarily mean efficient

The used reference system is uniform J with 8 creatures (J-8)
The most efficient systems

- Increase of efficiency by using time-shuffling?  
  (→ Yes, but limited)

Most of the efficient systems are uniform (J, B, C)

<table>
<thead>
<tr>
<th>No. of Creatures (k)</th>
<th>Algorithm X</th>
<th>Algorithm Y</th>
<th>Shuffle Mode</th>
<th>mean g_max</th>
<th>mean normalized work</th>
<th>absolute efficiency compared to J-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>J</td>
<td>J</td>
<td>u</td>
<td>382</td>
<td>12,74 €</td>
<td>1,184</td>
</tr>
<tr>
<td>28</td>
<td>B</td>
<td>C</td>
<td>c</td>
<td>437</td>
<td>12,74 €</td>
<td>1,182</td>
</tr>
<tr>
<td>16</td>
<td>J</td>
<td>J</td>
<td>u</td>
<td>768</td>
<td>12,80 €</td>
<td>1,176</td>
</tr>
<tr>
<td>28</td>
<td>B</td>
<td>B</td>
<td>u</td>
<td>456</td>
<td>13,30 €</td>
<td>1,132</td>
</tr>
<tr>
<td>12</td>
<td>J</td>
<td>J</td>
<td>u</td>
<td>1069</td>
<td>13,36 €</td>
<td>1,127</td>
</tr>
<tr>
<td>28</td>
<td>J</td>
<td>J</td>
<td>u</td>
<td>463</td>
<td>13,49 €</td>
<td>1,116</td>
</tr>
<tr>
<td>28</td>
<td>C</td>
<td>C</td>
<td>u</td>
<td>475</td>
<td>13,85 €</td>
<td>1,087</td>
</tr>
<tr>
<td>60</td>
<td>J</td>
<td>J</td>
<td>u</td>
<td>224</td>
<td>14,03 €</td>
<td>1,073</td>
</tr>
<tr>
<td>64</td>
<td>B</td>
<td>C</td>
<td>c</td>
<td>218</td>
<td>14,51 €</td>
<td>1,038</td>
</tr>
<tr>
<td>60</td>
<td>B</td>
<td>C</td>
<td>c</td>
<td>233</td>
<td>14,58 €</td>
<td>1,033</td>
</tr>
<tr>
<td>8</td>
<td>J</td>
<td>J</td>
<td>u</td>
<td>1807</td>
<td>15,06 €</td>
<td>1,000</td>
</tr>
</tbody>
</table>

- Increase of efficiency by increasing k?  
  (→ (Yes))

The top 10 most efficient systems use more than 12 creatures

- Efficiency higher than 1 means synergy through cooperation
Time-shuffle periods

- Generation-wise shuffling does not seem to improve the creatures' behavior if there are many.
- The new idea: toggle between behaviors after a certain amount of generations.
- Recent results: toggle after approx. 100 generations leads to a much better performance.

t=0/1 toggles every $T$ generations.
Non-uniform spatial

- Another way to mix algorithms: use creatures with different behavior in one system
- The initial placement of different creatures and the mixing ratio are interesting variables
- Recent results: performance improvements are comparable to time-shuffling
Conclusion

- Creature's Exploration Problem was modeled by CA
- Conflicts are solved using a new 1-phase arbitration logic
- Behavior of a creature is modeled by a Mealy-Control-Machine and an Action-Machine
- Three time-shuffling modes were proposed
- For one creature the time-shuffling modes leads to more success
- New metrics for multi-creature systems were defined

- More creatures lead to more success, to faster exploring and to more efficiency
- In most cases time-shuffling with many creatures does not lead to more efficiency (with time shuffle period T=1)

- Further investigations planned with
  - more sophisticated time-shuffling modes
  - different actions, (e.g. move forward, move backward)
  - communicating creatures
  - heuristic optimization methods
initial configuration (0 generations)
4 generations

Shades mark already visited cells
8 generations

Darker shades mark more often visited cells
16 generations
24 generations
41 generations
61 generations
67 generations
72 generations
93 generations
101 generations
119 generations
120 generations: success