Are several creatures more efficient than a single one?

Rolf Hoffmann and Mathias Halbach

- Creature's Exploration Problem
- CA Model
  - Cell Type
  - Actions of the Creature
  - Rule
  - Conflicts, 1-Phase Arbitration
  - Different Behaviors of a Creature
- Performance of the best algorithms for one creature
- Several Creatures
  - Cooperative work
  - Relative Efficiency
  - Absolute Efficiency
- Example with 64 creatures
- Conclusion
Creature's 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 by a robot
  - Exploring an unknown environment
The Actions of the Creatures

<table>
<thead>
<tr>
<th></th>
<th>(turn Left)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L</strong></td>
<td></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 showing the actions of creatures with arrows indicating movement and rotation.]
The Rule for the Creatures

(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)

Legend

- □ obstacle or creature
- □ irrelevant
- □ creature in one out of two directions
## CA: Cell Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Direction $r$</th>
<th>Control state $s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATURE</td>
<td>toNorth, toEast, toSouth, toWest</td>
<td>0, 1, ... (N-1)</td>
</tr>
<tr>
<td>EMPTY</td>
<td></td>
<td>(irrelevant)</td>
</tr>
<tr>
<td>OBSTACLE</td>
<td></td>
<td>(irrelevant)</td>
</tr>
</tbody>
</table>
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 **Output-Machine**
- Control Machine implements **State Algorithm**

grant signal from neighbor in front (front cell)

<table>
<thead>
<tr>
<th>m</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>2</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>0</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>4</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>5</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>3</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>R</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

s control state
r direction
d action
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)

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

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

- string representation

1L2L0L4R5R3R-3Lm1Rm5Lm0Rm4Lm2Rm
= 1L2L0L4R5R3R-3L1R5L0R4L2R

simplified
Performance of the best 6-state algorithms for one creature

The table shows the performance of different algorithms on a 26-state configuration (Test set). The algorithms are denoted as A6, B6, C6, D6, E6, F6, G6, H6, I6, J6, and K6. Each algorithm is evaluated for its success rate, percentage of successful crossings, and speed. The successful configurations are marked in blue, indicating 100% visited states, while the unsuccessful ones are marked in yellow.

- A6: 23 successful crossings, 99.07% success rate, 0.136 speed
- B6: 24 successful crossings, 99.92% success rate, 0.252 speed
- C6: 23 successful crossings, 99.29% success rate, 0.259 speed
- D6: 22 successful crossings, 98.63% success rate, 0.251 speed
- E6: 24 successful crossings, 99.12% success rate, 0.223 speed
- F6: 24 successful crossings, 97.94% success rate, 0.238 speed
- G6: 24 successful crossings, 99.92% success rate, 0.260 speed
- H6: 21 successful crossings, 98.91% success rate, 0.169 speed
- I6: 21 successful crossings, 98.91% success rate, 0.193 speed
- J6: 16 successful crossings, 78.55% success rate, 0.299 speed
- K6: 23 successful crossings, 99.87% success rate, 0.268 speed
The best algorithms for one creature

- G: 1L2L0L4R5R3R-3L1R5L0R4L2R
- B: 1R2R0R4L5L3L-3R1L5R0L4R2L
- C: 1R2R0R4L5L3L-3R4R2L0L1L5R
- A: 0R2R3R4L5L1L-1R5R4R0L2L3L
- D: 1R2R3R1L5L1L-1R0L2L4R3L1L
- E: 1R2L0R4L5L3L-3R4R5R0L1L2R
- F: 1R2L0L4R5R3R-3L4L5L0R1L2R
- H: 1L2L3R4L2R0L-2L4L0R3L5L4R
- I: 1L2L3L4L2R0L-2L4L0R3R5L4R
- J: 1R2R3R0R4L5L-4R5R3L2L0L1L
The two best 6-State-Algorithms

Algorithm G6
behaves slightly better for the 26 configs.

Algorithm B6
Several Creatures: Definitions

- $g_{\text{max}}(k) :=$ the number of generation needed to visit all cells by $k$ creatures

- The cooperative work of $k$ creatures is proportional to the number of generations (time steps) to visit all cells, multiplied with the number of creatures.
  - $W(k) = g_{\text{max}}(k) \times k$

- The relative efficiency is the work of one creature related to the work of $k$ creatures using the same algorithm $\text{Alg}$ for all the creatures.
  - $F_{\text{rel}} = W^{\text{Alg}}(1) / W^{\text{Alg}}(k)$

- The absolute efficiency is the work of one creature using the best algorithm $\text{AlgBest}$ divided by the work of $k$ creatures using the algorithm $\text{Alg}$.
  - $F_{\text{abs}} = W^{\text{AlgBest}}(1) / W^{\text{Alg}}(k)$
Example, Initial configurations

- Initial configurations used to evaluate the behavior of several creatures

\[ k = 1 \quad 4 \quad \ldots \quad 16 \quad \ldots \quad 64 \]
Generations $g_{max}$ to visit all empty cells

In most of the cases more creatures needed less time (generations), but in some cases more creatures need more time!!

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>1</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>647</td>
<td>718</td>
<td>375</td>
<td>196</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>3166</td>
<td>971</td>
<td>2012</td>
<td>476</td>
<td>332</td>
<td>66</td>
</tr>
<tr>
<td>C</td>
<td>3333</td>
<td>1120</td>
<td>629</td>
<td>1516</td>
<td>223</td>
<td>169</td>
</tr>
<tr>
<td>D</td>
<td>7525</td>
<td>1750</td>
<td>2080</td>
<td>763</td>
<td>508</td>
<td>64</td>
</tr>
<tr>
<td>E</td>
<td>4169</td>
<td>1120</td>
<td>1291</td>
<td>1544</td>
<td>587</td>
<td>58</td>
</tr>
<tr>
<td>F</td>
<td>4213</td>
<td>971</td>
<td>736</td>
<td>282</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>3166</td>
<td>971</td>
<td>2012</td>
<td>476</td>
<td>332</td>
<td>66</td>
</tr>
<tr>
<td>H</td>
<td>8009</td>
<td>1478</td>
<td>918</td>
<td>892</td>
<td>385</td>
<td>221</td>
</tr>
<tr>
<td>I</td>
<td>7168</td>
<td>1990</td>
<td>1777</td>
<td>449</td>
<td>296</td>
<td>44</td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td>4831</td>
<td>435</td>
<td>525</td>
<td>64</td>
</tr>
</tbody>
</table>

not successful (not all empty cells visited)
Question

- How many creatures and which algorithm should be used in order to complete the task with a minimum number of work (generation x creatures)?

MINIMUM: at least 2588 work units have to be paid using 4 creatures with algorithm A (647 generations).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Creatures</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2588</td>
</tr>
<tr>
<td>B</td>
<td>3884</td>
</tr>
<tr>
<td>C</td>
<td>5032</td>
</tr>
<tr>
<td>D</td>
<td>12208</td>
</tr>
<tr>
<td>E</td>
<td>24704</td>
</tr>
<tr>
<td>F</td>
<td>11776</td>
</tr>
<tr>
<td>G</td>
<td>16096</td>
</tr>
<tr>
<td>H</td>
<td>7344</td>
</tr>
<tr>
<td>I</td>
<td>14216</td>
</tr>
<tr>
<td>J</td>
<td>38648</td>
</tr>
</tbody>
</table>

But if you would spend about 9% more money (2816 instead of 2588), your task could be accomplished about 15 times faster (44 generations) using 64 creatures (I-64).
Relative Efficiency

- The **relative efficiency** is the work of one creature related to the work of $k$ creatures using the *same algorithm* $\text{Alg}$ for all the creatures.
  \[ F_{rel} = \frac{W^{\text{Alg}}(1)}{W^{\text{Alg}}(k)} \]

In the cases (D-64, I-64, H-4) the relative efficiency is greater than one. **This means that the work could be done cheaper using more creatures!!**
The absolute efficiency is the work of one creature using the best algorithm $\text{AlgBest}$ divided by the work of $k$ creatures using the algorithm $\text{Alg}$.

- $F_{\text{abs}} = \frac{W_{\text{AlgBest}}(1)}{W_{\text{Alg}}(k)}$

In the cases (A-4, I-64) the absolute efficiency is greater than one. This means that the work could really be done cheaper using more creatures!! $\rightarrow$ superlinear speed-up through cooperation.
Alg. I-64
generations 0,4,8,12,20,28,32,36,38,40,44

- The most efficient algorithm I with 64 creatures (I-64) was further investigated by simulation.
In each of the generations 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 38, 40, four conflicts are arising where two creatures are involved, in total 48 conflicts.

The sum of creatures who are involved in conflicts is in total: \((\text{conflicts}) \times 2 = 96\) which is very low, compared to the number of working units (2816).

Furthermore all conflicts, except for generation 38 have a positive effect because one of the creatures will visit an empty field in the next generation.

In this example the conflicts have a positive effect if the creatures meet each other collateral (angle 90°), whereas the conflicts are irrelevant if the creatures meet frontal (angle 180°).
Initial Configuration
4 conflicts
4 conflicts
4 conflicts
4 conflicts
4 conflicts
4 conflicts
4 conflicts
4 conflicts
4 conflicts
4 conflicts, not positive
4 conflicts
stop: all cells visited
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 Output-Machine
- In most cases more creatures can do the work faster, but not always (more creatures may do the work slower)
- The work can be done with a minimum of cost (work units) using the proper number of creatures and the proper algorithm
- Several creatures can perform the work more efficiently (cheaper) than only one
- **Superlinear Speedup** through cooperation (positive conflicts)
- Further investigations planned with creatures which
  - have different behaviors,
  - have different actions, (e.g. move forward, move backward)
  - can communicate with each other