Rectangular vs Triangular Routing with Evolved Agents

Patrick Ediger and Rolf Hoffmann
Technische Universität Darmstadt
Computer Architecture Group

Dominique Désérable
Institut National des Sciences Appliquées
Campus Universitaire de Beaulieu
35043 Rennes, France
Outline

I. Square grid (S-grid) and Triangulate grid (T-grid)
   • Diameter, Mean Distance

II. The Routing Task

III. Agents
   • Agent’s View, Moving, Behavior
   • Evolving the Behavior (FSMs)

IV. Comparison (S- vs T-grid)
   • Randomized Intelligent Walkers
   • Randomized FSMs

V. Special Case: One Common Target

VI. Conclusion
I. Square grid (S-grid) and Triangulate grid (T-grid)
Square Grid, S-grid

Torus
Triangulate Grid, T-grid

+ diagonal connections
Triangulate Grid, T-grid

- order $N = 2^n \times 2^n$, $n =$ size
- regular with degree 6
- Cayley graph: vertex transitive
Arrowhead Representation

$N = 16$

The **Arrowhead Torus** (Désérable 1999)

The associated dual tessellation of the plane is the regular **hexagonal tiling** (honeycomb)

Hexagonal representation of the arrowhead: $n = 2$, $N = 16$

Distances from center cell (in white) – Diameter 2
Distances in the S-grid and T-grid

$N = 64$

$Diameter = 8$

$Diameter = 5$
### S- and T-grid Diameter and Mean Distance

<table>
<thead>
<tr>
<th></th>
<th>Square-Grid</th>
<th>Triangulate-Grid</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter</td>
<td>$\sqrt{N}$</td>
<td>$\frac{2(\sqrt{N} - 1) + \varepsilon_n}{3}$</td>
<td>$\approx 1.5$</td>
</tr>
<tr>
<td>Mean Distance</td>
<td>$\frac{\sqrt{N}}{2}$</td>
<td>$\approx \frac{1}{6} \left( \frac{7\sqrt{N}}{3} - \frac{1}{\sqrt{N}} \right)$</td>
<td>$\approx 1.29$</td>
</tr>
</tbody>
</table>

\[ N = 2^n \times 2^n \]
\[ \varepsilon_n = 1/0 \text{ if } n \text{ odd/even} \]
II. The Routing Task
The Routing Task

*Find an optimal router for the S-grid using agents*

- **The Global Task:** All agents have to move from their initial source positions to their target positions
- **Goal**
  
  (1) Find the best behavior for the agents solving the global task in shortest time, for any initial configuration

  (2) Compare to the already known T-grid agents
The Routing Task
More Details

- celltype ∈ \{EMPTY, AGENT\}
- each agent moves from source cell to target cell
- each agent moves to a different target (assumed but not necessary)
- a target can also act as a source
- agent deletes itself when reaching its target
- **Static Routing**: All agents (messages) are injected initially at time $t = 0$, wait for the completion of all transfers (Barrier-synchronization)
- **Focus on the case**: $N = 1024$ cells, $k = N/2$ agents (one space per agent)
III. Agents
Agent’s View: 8 Target Areas

An agent can observe **8 target areas (colors)** where the target may be located.

The **view** of an agent is defined by the **input mapping**.

The **dashed targets** can be reached from two sides with the same distance; they are assigned to certain sectors to resolve this ambiguity.
Modeling Moving (0) Actions

• Agent has **4 directions**: N, E, S, W

• Basic Actions

  \[ mt_k : \text{move} \& \text{turn } k \times 90^\circ \]

  \[ st_k : \text{stay} \& \text{turn } k \times 90^\circ \]

• Conditional Actions (output from FSM)

  \[ T_k : \text{if (can move) then } mt_k \text{ else } st_k \]
Modeling the moving of an agent in CA requires a couple of two consistent rules:

(Receiver copies agent, Sender deletes agent)
Modeling Moving (2)

Swapping

*agents are swapping in order to avoid congestion and speeding up the task*
Conflict resolution has to be computed in the sending cell (S) and the receiving cell (R) based on the same information. A neighborhood of Manhattan-distance 2 in the agent’s direction is required in order to solve the conflict.
each cell comprises a priority scheme

(4! = 24 schemes, equally distributed over the cell space)
Modeling Behavior (1)

Cell Structure

- celltype
- direction
- own position
- target position
- priority

CONTROL UNIT

visible

embedded FSM
Modeling Behavior (2)
Control Unit (Embedded FSM)

- direction, target position, own position
- agent in front (B) OR conflicting agent (C)
- swap condition priority

CONTROL AUTOMATON
- 2 control inputs
- control state
- control outputs 4 conditional actions
- 8 basic actions
- move condition
- can move
- action mapping

8 control inputs (colors)
Modeling Behavior (3)

State Transition Table

- agent behaves according to the control algorithm, stored in a table
- 
- \((\text{state, input}) \rightarrow (\text{next state, conditional action})\)
- near optimal tables (algorithms) were evolved by a genetic procedure (Island genetic algorithm)
Evolving the Behavior
Island Model Genetic Procedure

- each run: **5 islands** with a **population of 100 genes (FSMs)**
- next state and output taken from either of the parents and mutated
- **fitness function**
  - # of successful solved initial configurations
  - # of not reached targets after 2 000 generations
  - # of generations to reach all targets (the time to transfer all messages)

(1) evolved on **Training Set** with 25 random initial configurations, 36 runs for CASE(N=16→64→256→1024)
  - **computation time** on Intel Xeon QuadCore 2 Ghz was **159 hours** altogether (1.55 times longer than for T-grid)

(2) the Top 10 of each run (360 algorithms) were **ranked by simulation on a Test Set**
  - final list of <360 successful algorithms
II. Comparison
(S-grid vs T-grid)
Additional Agents for Comparison: Randomized Intelligent Walkers IW(p)

• **Intelligent Walker IW**: tries to move **directly to the target**
  - if there are 2 equivalent alternatives: select one at random
• **Problem**: **Deadlocks** and **Livelocks**
• **Solution**: *randomness* $p$ *was added*: IW(p)
  - with $p$: turn randomly to any of the possible directions

• Optimal randomness was found by variation:
  - $N = 1024$, **S-grid** $\rightarrow p = 6\%$
  - $N = 1024$, **T-grid** $\rightarrow p = 12\%$
Randomized FSM(p)

- In order to avoid deadlocks with FSMs, the FSM agents were also randomized
  
  $FSM \rightarrow FSM(p)$

- obey to the FSM or choose a random direction with probability $p = 0.3\%$ (lower rate was used because deadlocks appear with lower frequency)
Simulations
Best Evolved Algorithms for the S-grid and T-grid

$T = 0$  

$48$  

$78$  

$100$  

$T = 0$  

$28$  

$42$  

$56$  

$107$  

$60$
Comparison
Delivered Messages for N = 1024 cells, 512 agents

Evolved FSM agents on S-grid are 1.91 x slower compared to T-grid

Randomized Intelligent Walkers IW are ≈1.15 x slower than Evolved Agents on T-grid
V. Special Case: One Common Target
Special Initial Placement
One Common Target

- One **common target (in the middle)** is used for all the agents
- Initially the agents are clustered around the target.

**S-grid**

Best Evolved Algorithm: Time Steps $\approx 517$

**T-grid**

Best Evolved Algorithm: Time Steps $\approx 514$

(averaged over initial configurations, averaged over diff. runs)
Simulation: One Common Target
S-grid and T-grid
Conclusion

• We solved the **routing problem** with \( \frac{N}{2} \) agents walking in a **rectangular grid**

• The agent’s behavior was **evolved**, and further **randomized**

• For comparison **Randomized Intelligent Walkers IW(p)** were defined
  - try to move directly
  - turn randomly with \( p \)
  - optimal \( p \) was used

• **Square vs Triangulate Grid, Evolved Agents:**
  \( \rightarrow 1.91 \times \) slower
  - (diameter 1.5 \( x \) larger)

• **Square Grid:**
  \( \rightarrow \) The best evolved FSM(p) agents is \( \approx 15\% \) faster than the IW(p)

• **Future Work**
  \( \rightarrow \) Compare with other grids
  \( \rightarrow \) Use more buffers per cell
Thank you very much for your Attention!