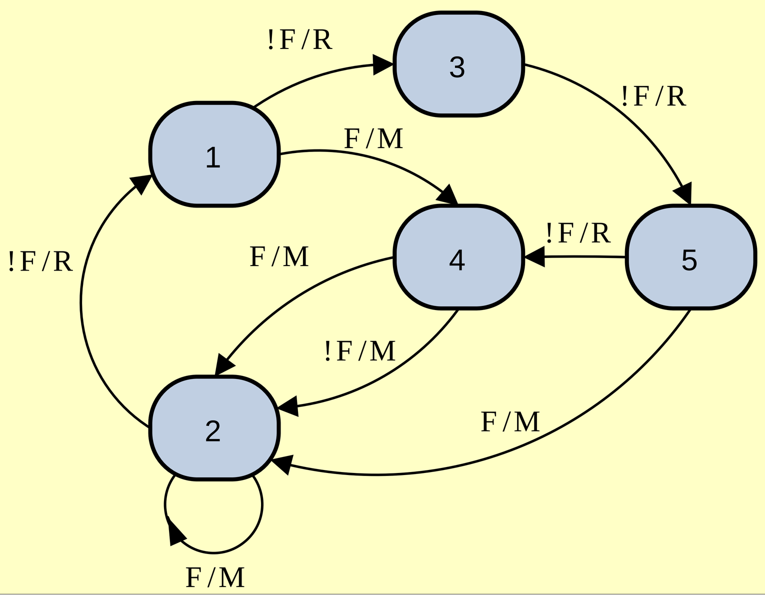
Solving Five Instances of the Artificial Ant Problem with Ant Colony Optimization

Daniil Chivilikhin, Vladimir Ulyantsev, Anatoly Shalyto
St. Petersburg National Research University of Information Technologies, Mechanics and Optics
Computer Technologies Department

Finite-State Machine

- S – set of states
- Σ – set of input events
- Δ – set of output actions
- $\delta : S \times \Sigma \to S$ – transition function
- $\lambda : S \times \Sigma \to \Delta$ – action function
- $s_0 \in S$ – initial state

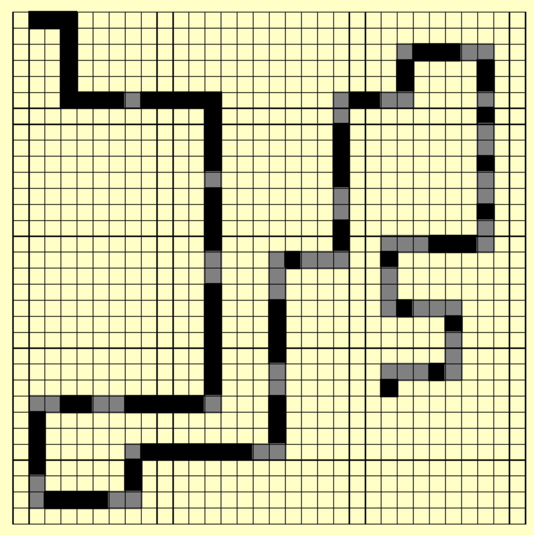


Artificial Ant Problem

- Two events: *F* (food ahead) and *!F* (no food ahead)
- Three actions: *L* (turn left), *R* (turn right), *M* (move forward)
- A maximum of s_{max} time steps
- Maximize fitness function:

$$f = n_{food} + \frac{s_{max} - s_{last} - 1}{s_{max}}$$

- n_{food} – eaten food count
- s_{last} – last step

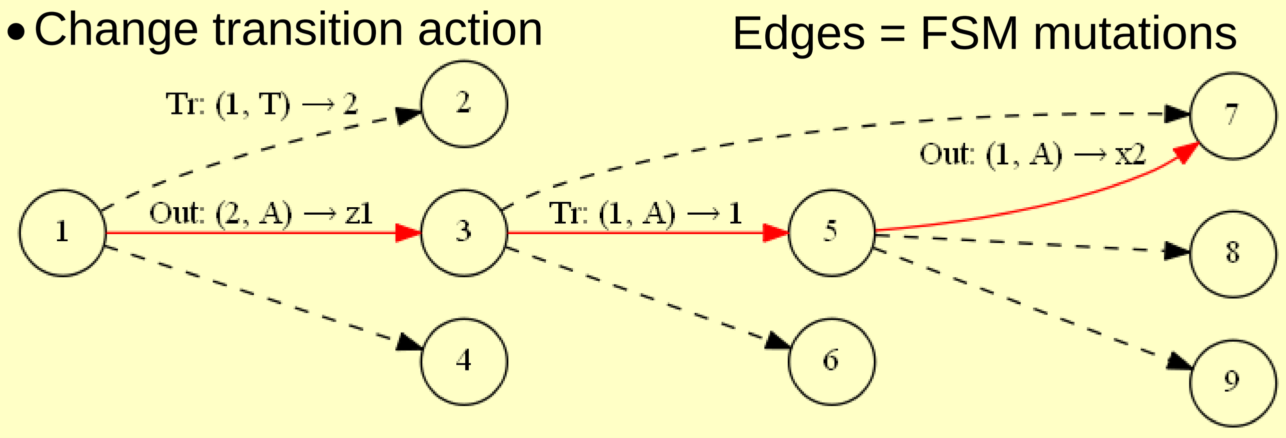


Santa Fe food trail field

Mutation-Based Ant Colony Optimization for Learning Finite-State Machines

FSM mutations:
- Change transition end state
- Change transition action

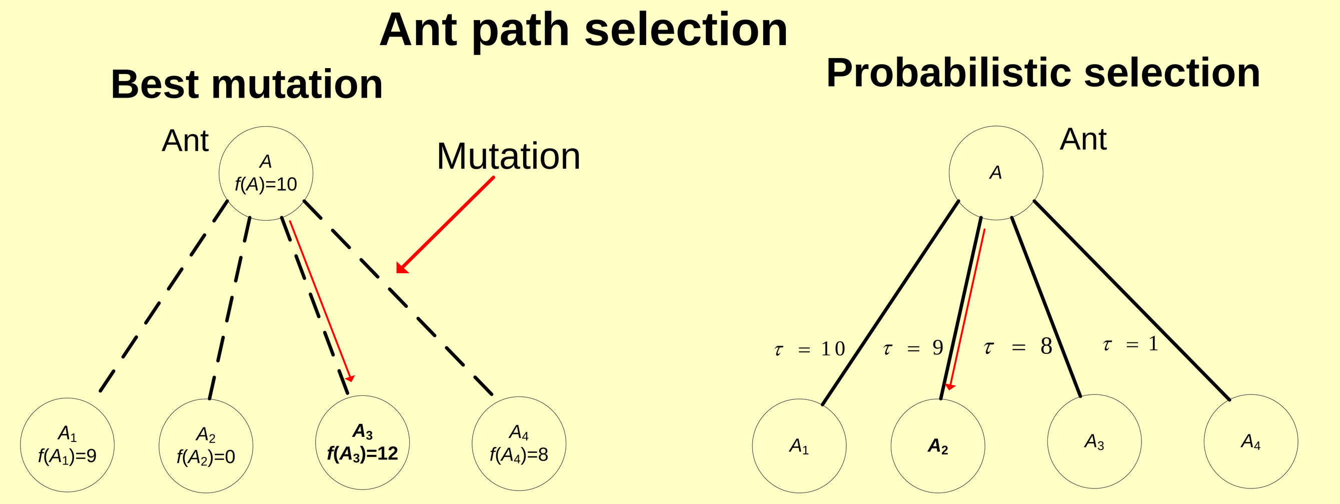
Construction graph:
Nodes = Finite-State Machines
Edges = FSM mutations



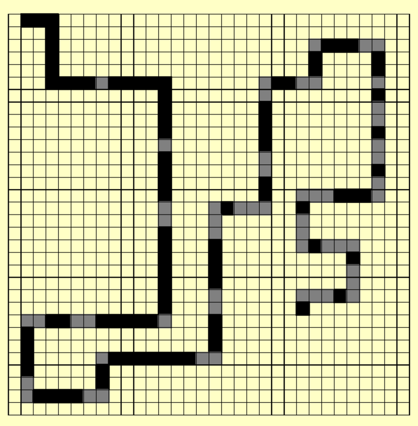
Algorithm

while (true)
  Build solutions with ant colony
  Update pheromone
  Check stop criteria

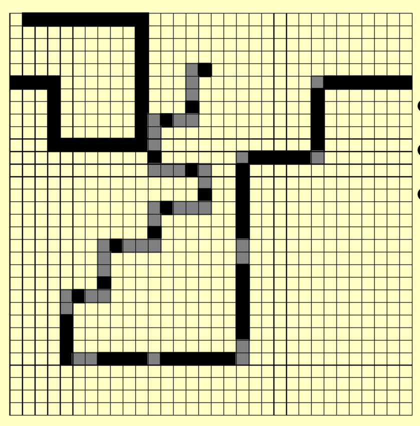
Ant path selection



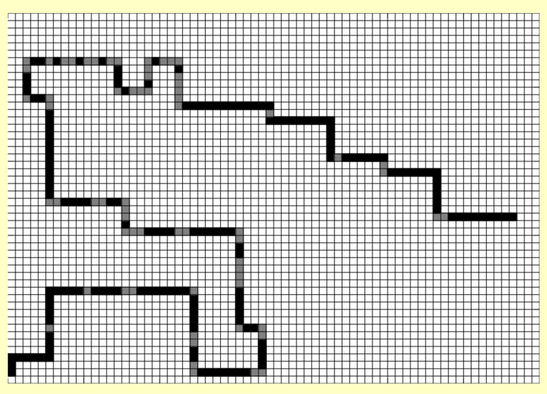
Experimental Results: Artificial Ant Problem



Santa Fe Trail
- s_{max} = 400, 500, 600
- Compare with Christensen et al. (2004)
- Measure mean fitness evaluations

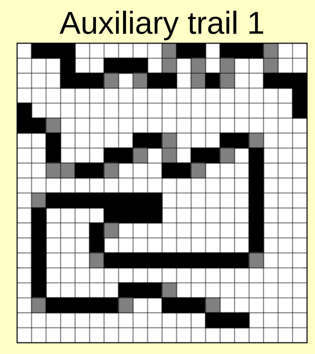


John Muir Trail
- s_{max} = 200
- Compare with Tsarev et al. (2007)
- Measure mean fitness evaluations



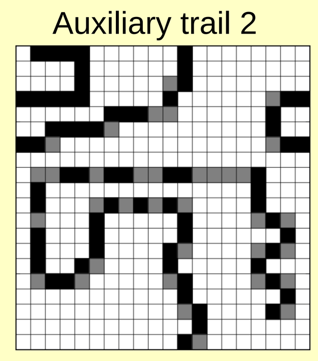
Los Altos Hills Trail
- s_{max} = 800
- Compare with Mesot et al. (2002)
- Measure mean number of ant steps

Auxiliary trail 1

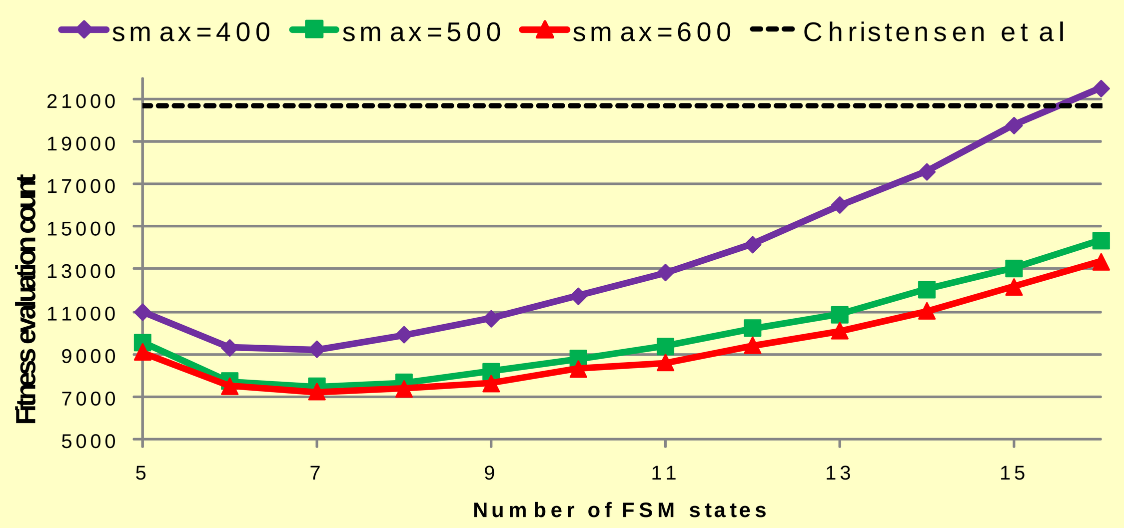


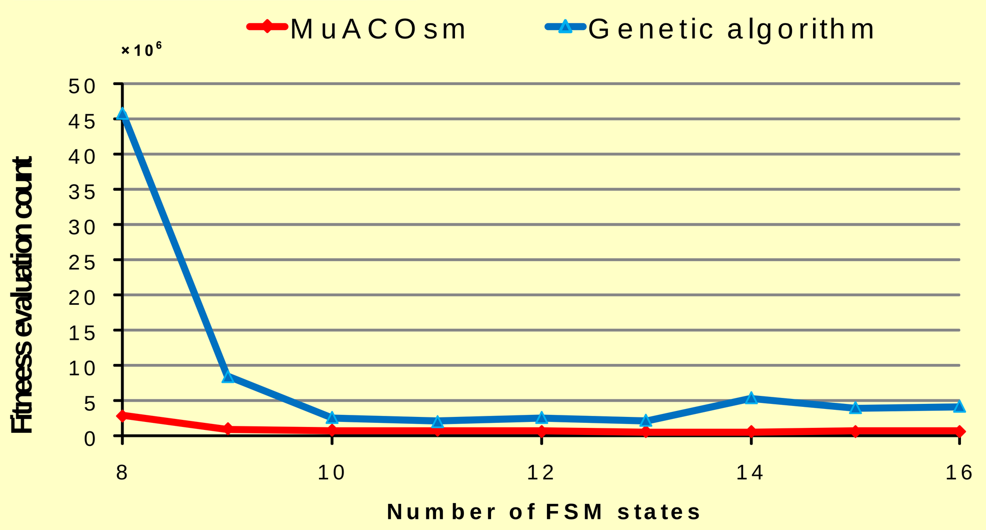
- MuACOsm is five times faster than evolution strategy

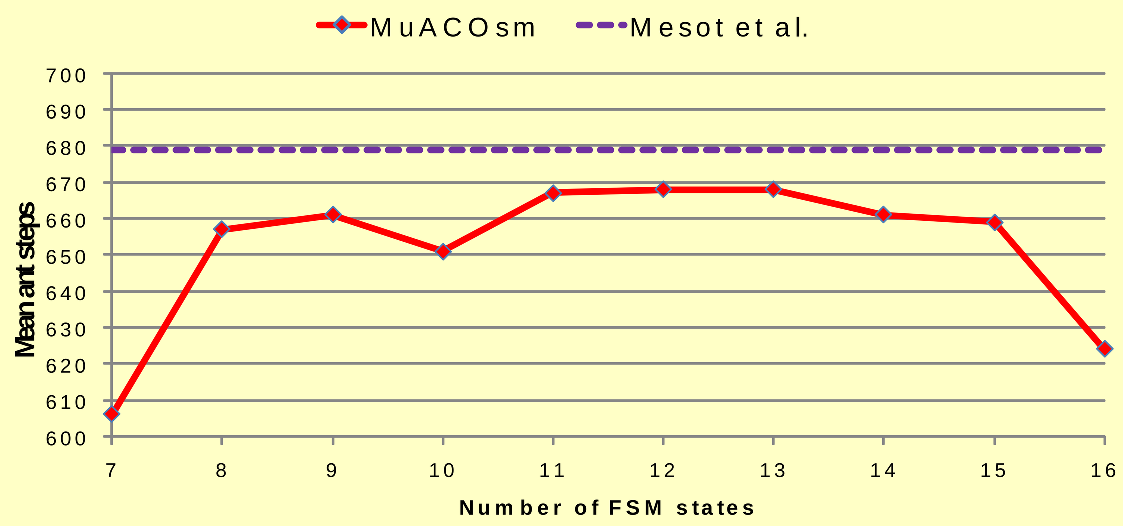
Auxiliary trail 2



- MuACOsm is six times faster than evolution strategy

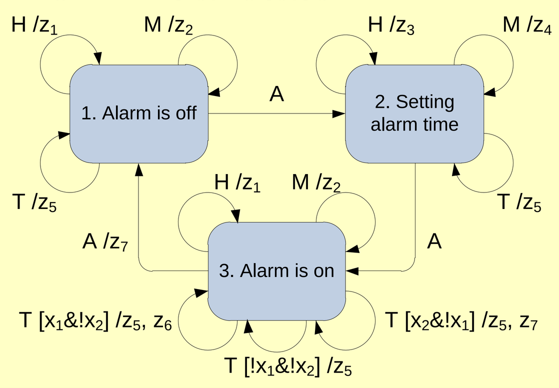






Experimental Results: Inducing EFSMs from test examples

Extended FSM:
- S – set of states
- Σ – set of input events
- Δ – set of output actions
- X – set of Boolean input variables
- $\delta : \Sigma$□E□$2^X \to S$ – transition function
- $\lambda : \Sigma$□E□$2^X \to Z^*$ – action function
- $s_0 \in S$ – initial state



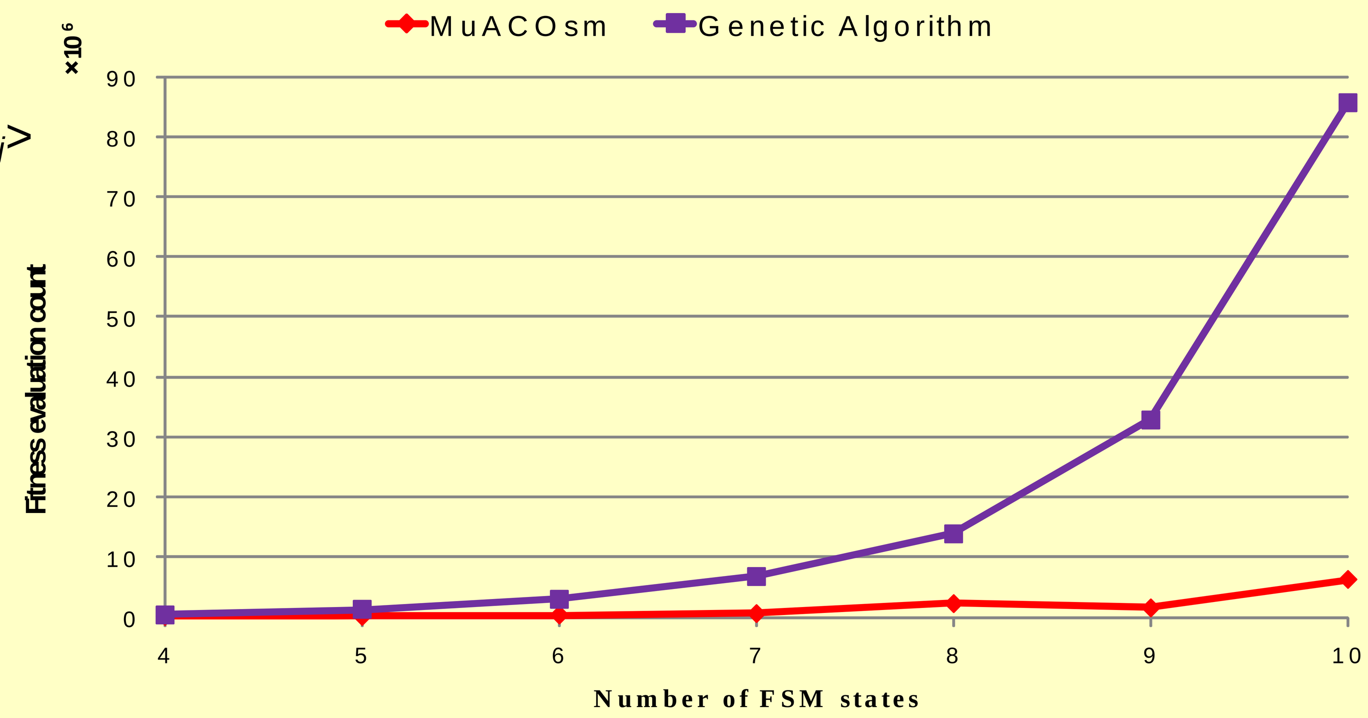
Input data:
- Number of states N and sets Σ and Δ
- Set of test examples T
- T_i=<events sequence I_j, output sequence O_j>

Set of tests → Induction Algorithm → EFSM

Fitness function:

$$f' = \frac{1}{|T|}\sum_{j=1}^{|T|}\left(1 - \frac{ED(O_j, A_j)}{\max\left(len(O_j), len(A_j)\right)}\right)$$

$$f = 100 \cdot f' + \frac{1}{100} \cdot (100 - n_{trans})$$



Experimental setup:
- 2 input events
- 2 output actions
- Max output actions = 2
- One input variable
- 100 experiments
- Tests size = $150 \cdot N_{states}$

Publications

- Chivilikhin D., Ulyantsev V. Learning Finite-State Machines with Ant Colony Optimization // Lecture Notes in Computer Science, 2012, Volume 7461/2012, pp. 68-275
- Chivilikhin D., Ulyantsev V., Tsarev F. Test-Based Extended Finite-State Machines Induction with Evolutionary Algorithms and Ant Colony Optimization / Proceedings of the 2012 GECCO Conference Companion on Genetic and Evolutionary Computation. NY.: ACM. 2012, pp. 603 – 606.
- Ulyantsev V., Tsarev F. Extended Finite-State Machine Induction using SAT-Solver / Proceedings of the Tenth International Conference on Machine Learning and Applications, ICMLA 2011, Honolulu, HI, USA, 18-21 December 2011. IEEE Computer Society, 2011. Vol. 2. P. 346–349.

Summary

- MuACOsm outperforms all published algorithms on inducing FSMs for Artificial Ant Problem
- MuACOsm significantly outperforms GA on inducing EFSMs from test examples

chivilikhin.daniil@gmail.com
ulyantsev@rain.ifmo.ru