Test-Based Induction of Finite-State Machines with Continuous Output Actions
Igor Buzhinsky, Vladimir Ulyantsev, Anatoly Shalyto
St. Petersburg National Research University of Information Technologies, Mechanics and Optics
Computer Technologies Department

Problem Statement

Finite-State Machine:
- FSM = (S, Σ, Δ, δ, λ, s₀)
- S – finite set of states
- Σ, Δ – event and action sets
- δ: S × E → S – transition function
- λ: S × E → Δ – output function
- s₀ – start state

Problem:
- Control object has continuous (real) control parameters
- Tests are the examples of proper control
- Given a set of N tests (N = 20–30), an FSM should be constructed with behavior close to the tests
- Aircraft model is used as a control object
- Tests can be written manually in a flight simulator

Method Summary

- Ant colony optimization (ACO) and evolution strategy (ES) are used instead of genetic algorithm (GA) which was used earlier for the same problem
- Predicates transform continuous inputs to discrete events
- FSMs without actions (FSM skeletons) are ACO / ES individuals
- Actions are assigned so that fitness function is maximized:

Experiments & Results

- Compared ACO, (μ, λ)-ES and GA
- Intel Core 2 Quad Q9400 processor, four cores
- 25 runs of ACO / ES / GA for each test set
- Searching for FSMs with four states
- Numbers of runs in which the fitness values were reached:

<table>
<thead>
<tr>
<th>Fitness value</th>
<th>ACO</th>
<th>(μ, λ)-ES</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9890</td>
<td>11</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>0.9987</td>
<td>21</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>0.9984</td>
<td>24</td>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>0.9981</td>
<td>24</td>
<td>24</td>
<td>17</td>
</tr>
<tr>
<td>0.9978</td>
<td>24</td>
<td>24</td>
<td>21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fitness value</th>
<th>ACO</th>
<th>(μ, λ)-ES</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9884</td>
<td>8</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0.9882</td>
<td>23</td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>0.9980</td>
<td>25</td>
<td>24</td>
<td>15</td>
</tr>
<tr>
<td>0.9878</td>
<td>25</td>
<td>24</td>
<td>19</td>
</tr>
<tr>
<td>0.9976</td>
<td>25</td>
<td>24</td>
<td>24</td>
</tr>
</tbody>
</table>

- ACO and (μ, λ)-ES outperform GA, ACO slightly outperforms ES
- Run time ≈ 20 minutes
- About 90% of generated FSMs were able to perform the aerobatic figures

Input Data: Tests

- in[λ][i][j] – inputs (flight parameter values)
- out[λ][i][j] – outputs (control parameter values)

Test example

<table>
<thead>
<tr>
<th>Values</th>
<th>Meaning</th>
<th>t = 1</th>
<th>...</th>
<th>t = 235</th>
</tr>
</thead>
<tbody>
<tr>
<td>in[λ][i][1]</td>
<td>Pitch angle</td>
<td>3.078</td>
<td>...</td>
<td>4.112</td>
</tr>
<tr>
<td>in[λ][i][2]</td>
<td>Airspeed (knots)</td>
<td>251.42</td>
<td>...</td>
<td>253.20</td>
</tr>
<tr>
<td>out[λ][i][1]</td>
<td>Aileron position</td>
<td>0.000</td>
<td>...</td>
<td>0.073</td>
</tr>
<tr>
<td>out[λ][i][2]</td>
<td>Elevator position</td>
<td>-0.035</td>
<td>...</td>
<td>-0.037</td>
</tr>
</tbody>
</table>

Screenshots (FlightGear)

- Loop test set (trajectories)
- Barrel roll test set (trajectories)

Ant Colony Optimization

- Construction graph: vertices – FSM skeletons, edges – mutations (small changes in skeletons)
- Initially, graph consists of a single randomly generated vertex / skeleton
- Graph grows during algorithm execution
- Ants wander using pheromone

Publications

- Ulyantsev V., Tsarev F. Extended Finite-State Machine Induction using SAT-Solver / Proceedings of the 14th IFAC Symposium "Information Control Problems in Manufacturing - INCOM'12", IFAC, 2012, pp. 512-517

(buzhinsky, ulyantsev)@rain.ifmo.ru

MIM 2013