Inferring Automata-Based Programs from Specification With Mutation-Based Ant Colony Optimization (Supplementary Material)

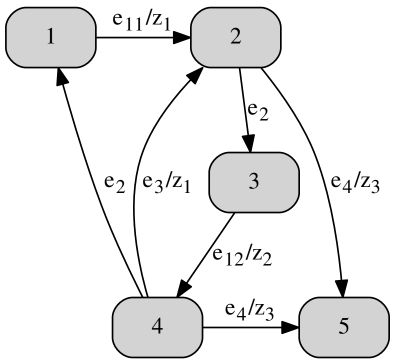

Figure 1: Elevator doors control EFSM

1. ELEVATOR DOORS PROBLEM

In this problem the controlled object is a model of the doors of an elevator. The problem is characterized by five input events:

- e_{11} – user pressed button “Open the doors”;
- e_{12} – user pressed button “Close the doors”;
- e_2 – doors successfully opened;
- e_3 – some obstacle prevented doors from closing;
- e_4 – doors are jammed.

The corresponding three output actions are:

- z_1 – start opening doors;
- z_2 – start closing doors;
- z_3 – call the emergency.

Table 1 contains examples of all types of input data. The target EFSM with five states is shown in Fig. 1.

Table 1: Examples of test scenarios and LTL formulae

Example	Description
Test scenarios	
$e_{11}/z_1; e_2; e_{12}/z_2; e_4/z_3$	Doors jammed after closing: we explicitly see that event e_2 does not require any action
$e_{11}/z_1; e_2; e_{12}/z_2; e_3/z_1; e_2; e_{12}/z_2; e_2$	Obstacle prevented doors from closing, closed at second try
LTL formulae	
$G(wasEvent\,(e_3) \implies wasAction\,(z_1))$	If an obstacle an obstacle prevents the doors from closing, they will start opening
$G(wasAction\,(z_1) \implies X[U(!wasAction(z_1), wasAction(z_2) \mid\mid wasEvent(e_4))])$	If doors started opening, they will not start to open again until they close or jam

2. ALARM CLOCK PROBLEM

Here the controlled object is an alarm clock which has three buttons: “H” (hours), “M” (minutes) and “A” (alarm). Buttons are used for setting the current time on the clock, setting alarm time and turning it on or off. The clock has two basic modes of operation for setting the current time and for setting the alarm time. In the “Setting current time” mode the alarm is off and the user can use buttons “H” and “M” for setting current time. When the user presses the “A” button in this mode, the clock switches to “Setting alarm time” mode, where the same buttons “H” and “M” are used to set the alarm time. If the user presses the “A” button in this mode, the alarm is set to go off at the selected time. While the alarm is ringing, the user can use the “A” button to switch it off, however it will be switched off automatically after one minute. The alarm clock also has a timer that each minute increases the current time by one minute.

The system is characterized by four input events: H (the “H” button has been pressed), M (the “M” button has been pressed), A (the “A” button has been pressed) and T (timer event). There are also two input variables:

- x_1 – is the alarm time equal to the current time?

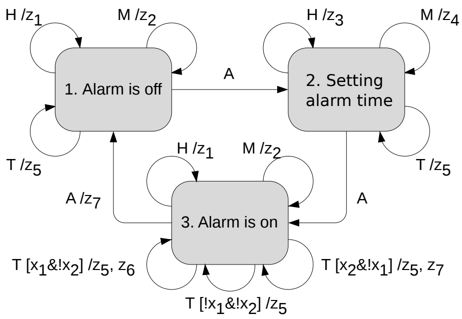

Figure 2: Alarm clock control EFSM

- x_2 – is the current time one minute greater than the alarm time?

The system also has seven output actions:

- z_1 – increase the number of hours of the current time by one;
- z_2 – increase the number of minutes of the current time by one;
- z_3 – increase the number of hours of the alarm time by one;
- z_4 – increase the number of minutes of the alarm time by one;
- z_5 – increase current time by one minute;
- z_6 – turn the alarm on;
- z_7 – turn the alarm off.

The EFSM that controls the alarm clock is shown in Fig. 2.

3. PARAMETER TUNING

When generating the problem instance set for tuning we used the following settings: the number of states N_{states} in the generated EFSM was selected randomly between 4 and 6, $5 \times N_{\text{states}}$ test scenarios with a total length of $100 \times N_{\text{states}}$ were generated for each EFSM. Generated EFSMs had 3 input events, two input variables, two output actions, the length of output sequences varied from 0 to 3. During tuning, algorithms were run for no longer than 10 seconds on each problem instance and compared based on reached fitness function values.

Parameter values selected using `irace` are shown in Table 2 for GA and Table 3 for MuACO*sm*. Parameter names are given as they appear in configuration files for GA and MuACO*sm*.

Table 2: Parameter ranges and selected parameter values for GA

Parameter	Range	Value
populationSize	$[100, 2000]$	137
partStay	$[0.01, 0.5]$	0.47
mutationProbability	$[0.01, 0.1]$	0.08
timeSmallMutation	$[50, 500]$	286
timeBigMutation	$[100, 1000]$	255

Table 3: Parameter ranges and selected parameter values for MuACO*sm*

Parameter	Range	Value
number-of-ants	$[1, 20]$	1
number-of-mutations-per-step	$[1, 50]$	44
big-stagnation-parameter	$[2, 50]$	15
stagnation-parameter	$[2, 50]$	22
evaporation-rate	$[0.1, 0.9]$	0.44