Road Traffic Rules Synthesis using Grammatical Evolution

Eric Medvet    Alberto Bartoli    Jacopo Talamini

Department of Engineering and Architecture
University of Trieste
Italy

EvoAPPS, 19/4/2017, Amsterdam (The Netherlands)

http://machinelearning.inginf.units.it
Driving

Human driver (now)

Are those rules “optimal” for both?
Driving

Human driver (*now*)

Driverless car (*the future*)

Are those rules “optimal” for both?
Driving

Human driver (*now*)

Driverless car (*the future*)

Both try to:
- get there
- avoid accidents
- comply to the rules
Driving

Human driver (*now*)

Driverless car (*the future*)

Both try to:

- get there
- avoid accidents
- comply to the rules

Are those rules “optimal” for both?

Medvet et al. (UniTs)
If we could re-write the rules from scratch, we’d want the rules leading to
- high traffic efficiency (getting there) and
- high safety (avoiding accidents)

How to write those rules?
If we could re-write the rules from scratch, we’d want the rules leading to
- high traffic efficiency (getting there) and
- high safety (avoiding accidents)

How to write those rules?

Our solution: write the rules automatically with an evolutionary approach
In a nutshell

Evolutionary approach:
- individual $\rightarrow$ set of rules
- fitness $\rightarrow$ $\langle$efficiency, safety$\rangle$, simulated with those rules

Our contributions:
1. a model for traffic simulation, suitable for our scenario
2. a language for defining rules applicable to the model
3. a GE-based framework for automatic rules generation, with experimental evaluation
In a nutshell

Evolutionary approach:
- individual $\rightarrow$ set of rules
- fitness $\rightarrow$ $\langle$efficiency, safety$\rangle$, simulated with those rules

Our contributions:
1. a *model* for traffic simulation, suitable for our scenario
2. a *language* for defining rules applicable to the model
3. a *GE-based framework* for automatic rules generation, with experimental evaluation
# Table of Contents

1. The traffic model
2. Language for rules
3. GE-based rules generation
4. Experimental evaluation
The traffic model

Goals (and trade-offs):
- detailed enough to include concepts amenable to be regulated
- simple enough to allow for simulation

Describes:
- the physical world
- the drivers’ behavior
The traffic model

Physical world

Infrastructure:
- road sections and intersections
- lanes in sections

Cars:
- position (continuous longitudinally, discrete on lanes)
- speed
- status (in \{alive, dead\})

Collisions

With discrete time
The traffic model

Drivers’ behavior

The driver’s algorithm:

- **input:** view ahead, speeds, distance to travel
- **output:** a list \( A \) of actions
  
  - \( \uparrow \): accelerate
  - \( \leftarrow \): move on left lane
  - … and so on (\( \uparrow, \uparrow, \rightarrow, \downarrow, \checkmark, \leftarrow, \checkmark \))
Drivers’ behavior

Goal:
- just travel a target distance, no specific target position
- possibly at maximum (car) speed
- avoiding hitting other cars
Drivers’ behavior and rules

The rules-aware driver’s algorithm:
- input: a list $A$ of actions and a set $R$ of rules
- output: the first action $a \in A$ which breaks the lowest number of rules
Drivers’ behavior and rules

The rules-aware driver’s algorithm:
- input: a list $A$ of actions and a set $R$ of rules
- output: the first action $a \in A$ which breaks the lowest number of rules

Why driver and rules-aware driver?
- model the driver’s “instinct” to perform his favorite action
- with no rules: “I drive as I like”
- with rules: “among preferred actions, I choose a permitted one”
Table of Contents

1. The traffic model
2. Language for rules
3. GE-based rules generation
4. Experimental evaluation
Rules

Rule:
- each rule is a predicate
- at a given time, for a given rule, a car breaks (false) or does not break (true) the rule

Works on:
- the car status (speed, position in section/intersection, ...)
- other cars relative distances and speeds
- current section/intersection
Language for the rules

As a context-free grammar:

\[ r ::= \langle \text{conditions} \rangle \]
\[ \langle \text{conditions} \rangle ::= \langle \text{condition} \rangle \mid \langle \text{conditions} \rangle \lor \langle \text{condition} \rangle \]
\[ \langle \text{condition} \rangle ::= \langle \text{baseCondition} \rangle \mid \neg \langle \text{baseCondition} \rangle \]
\[ \langle \text{baseCondition} \rangle ::= \langle \text{numericCondition} \rangle \mid \langle \text{deltaCondition} \rangle \mid \langle \text{graphCondition} \rangle \]
\[ \langle \text{numericCondition} \rangle ::= \langle \text{numericVariable} \rangle \leq \langle \text{numericValue} \rangle \]
\[ \langle \text{numericVariable} \rangle ::= \hat{x} \mid v_{\text{max}} \mid v_{\Delta} \mid d_{\text{view}} \mid d_{\epsilon} \mid \hat{x} \mid \hat{y} \mid \Delta x_{-1} \mid \Delta x_{0} \mid \Delta x_{1} \mid l(p) \mid w(p) \]
\[ \langle \text{numericValue} \rangle ::= \langle \text{digit} \rangle . \langle \text{digit} \rangle \text{e} \langle \exp \rangle \]
\[ \langle \text{digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \]
\[ \langle \exp \rangle ::= -1 \mid 0 \mid 1 \]
\[ \langle \text{deltaCondition} \rangle ::= \delta v_{-1} = \langle \text{deltaValue} \rangle \mid \delta v_{0} = \langle \text{deltaValue} \rangle \mid \delta v_{1} = \langle \text{deltaValue} \rangle \]
\[ \langle \text{deltaValue} \rangle ::= \emptyset \mid \text{opposite} \mid -1 \mid 0 \mid 1 \]
\[ \langle \text{graphCondition} \rangle ::= p \in S \]
Language for the rules

As a context-free grammar:

\[
\begin{align*}
    r & ::= \langle \text{conditions} \rangle \\
    \langle \text{conditions} \rangle & ::= \langle \text{condition} \rangle \mid \langle \text{conditions} \rangle \lor \langle \text{condition} \rangle \\
    \langle \text{condition} \rangle & ::= \langle \text{baseCondition} \rangle \mid \neg \langle \text{baseCondition} \rangle \\
    \langle \text{baseCondition} \rangle & ::= \langle \text{numericCondition} \rangle \mid \langle \text{deltaCondition} \rangle \mid \langle \text{graphCondition} \rangle \\
    \langle \text{numericCondition} \rangle & ::= \langle \text{numericVariable} \rangle \leq \langle \text{numericValue} \rangle \\
    \langle \text{numericVariable} \rangle & ::= \hat{v}_x \mid v_{\text{max}} \mid v_{\Delta} \mid d_{\text{view}} \mid d_\epsilon \mid \hat{x} \mid \hat{y} \mid \Delta x_{-1} \mid \Delta x_0 \mid \Delta x_1 \mid l(p) \mid w(p) \\
    \langle \text{numericValue} \rangle & ::= \langle \text{digit} \rangle . \langle \text{digit} \rangle ^{\langle \text{exp} \rangle} \\
    \langle \text{digit} \rangle & ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\
    \langle \text{exp} \rangle & ::= -1 \mid 0 \mid 1 \\
    \langle \text{deltaCondition} \rangle & ::= \delta v_{-1} = \langle \text{deltaValue} \rangle \mid \delta v_0 = \langle \text{deltaValue} \rangle \mid \delta v_1 = \langle \text{deltaValue} \rangle \\
    \langle \text{deltaValue} \rangle & ::= \emptyset \mid \text{opposite} \mid -1 \mid 0 \mid 1 \\
    \langle \text{graphCondition} \rangle & ::= p \in S
\end{align*}
\]

Goal:

* express realistic rule like “stay on rightmost free lane”
Table of Contents

1. The traffic model
2. Language for rules
3. GE-based rules generation
4. Experimental evaluation
We have:

- the fitness (traffic efficiency and safety)
- the solution space (as a context-free grammar)

⇓

GE is the right tool!
Fitness

- Traffic efficiency

- Traffic safety
Fitness

- Traffic efficiency $\rightarrow$ average speed ratio (ASR)
  \[
  \frac{1}{n_{\text{sim}}} \frac{1}{n_{\text{car}}} \sum_{\text{cars}} \left(1 - \frac{d_{\text{tot}}}{k_{\text{tot}}} \frac{1}{v_{\text{max}}} \right)
  \]

- Traffic safety $\rightarrow$ $1 -$ collision-per-time (CpT)
  \[
  \frac{1}{n_{\text{sim}}} \frac{1}{n_{\text{car}}} \sum_{\text{cars}} \frac{n_{\text{collision}}}{k_{\text{tot}}}
  \]

Minimize a linear combination: $f(R) = \alpha_{\text{time}} (1 - \text{ASR}) + \alpha_{\text{collision}} \text{CpT}$
Fitness

- Traffic efficiency → average speed ratio (ASR)

\[ \frac{1}{n_{\text{sim}}} \frac{1}{n_{\text{car}}} \sum_{\text{cars}} \left( 1 - \frac{d_{\text{tot}}}{k_{\text{tot}}} \frac{1}{v_{\text{max}}} \right) \]

- Traffic safety → 1 – collision-per-time (CpT)

\[ \frac{1}{n_{\text{sim}}} \frac{1}{n_{\text{car}}} \sum_{\text{cars}} \frac{n_{\text{collision}}}{k_{\text{tot}}} \]

Minimize a linear combination: \( f(R) = \alpha_{\text{time}} (1 - \text{ASR}) + \alpha_{\text{collision}} \text{CpT} \)

Computed over many \( (n_{\text{sim}}) \) simulations (stochasticity)
Table of Contents

1 The traffic model
2 Language for rules
3 GE-based rules generation
4 Experimental evaluation
Experimental evaluation

Goals

Traffic model:
- is it sound?
- find values for parameters
Goals

Traffic model:
- is it sound?
- find values for parameters

GE-based rules generation:
- are generated rules better than no rules?
- are generated rules better than hand-written rules?
Experimental evaluation

Goals

Traffic model:
- is it sound?
- find values for parameters

GE-based rules generation:
- are generated rules better than no rules?
- are generated rules better than hand-written rules?

For both: how does injected traffic affect efficiency and safety?
- congestion: no further increase in overall traveled distance
Traffic model validation

5 road sections, up to tens of cars, values averaged on 10 simulations

\[ n_{\text{car}} \text{ (injected traffic)} \]

1 − ASR (Efficiency)

Low efficiency

High efficiency

CpT (Safety)

Low safety

High safety

\[ n_{\text{car}} \text{ (injected traffic)} \]
Traffic model validation

5 road sections, up to tens of cars, values averaged on 10 simulations

\[ 1 - \text{ASR (Efficiency)} \]

\[ \cdot10^{-2} \]

\[ \frac{\text{CpT (Safety)}}{\text{Efficiency}} \]

\( n_{\text{car}} \) (injected traffic)

No rules

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Safety</th>
</tr>
</thead>
<tbody>
<tr>
<td>low ( n_{\text{car}} )</td>
<td>high</td>
</tr>
<tr>
<td>high ( n_{\text{car}} )</td>
<td>low</td>
</tr>
</tbody>
</table>
Traffic model validation

5 road sections, up to tens of cars, values averaged on 10 simulations

- **1 - ASR (Efficiency)**
  - No rules: low $n_{\text{car}}$ vs. high $n_{\text{car}}$ efficiency
  - Hand-written rules: low $n_{\text{car}}$ vs. high $n_{\text{car}}$ efficiency

- **CpT (Safety)**
  - No rules: high efficiency vs. high safety
  - Hand-written rules: low efficiency vs. low safety
Traffic model validation

5 road sections, up to tens of cars, values averaged on 10 simulations

\[ 1 - \text{ASR (Efficiency)} \]

\[ \cdot 10^{-2} \]

\[ n_{\text{car}} \text{ (injected traffic)} \]

\[ n_{\text{car}} \text{ (injected traffic)} \]

No rules

\[ \begin{array}{ccc}
\text{Efficiency} & \text{Safety} \\
\text{low} n_{\text{car}} & \text{high} & \text{high} \\
\text{high} n_{\text{car}} & \text{low} & \text{low}
\end{array} \]

Hand-written rules

\[ \begin{array}{ccc}
\text{Efficiency} & \text{Safety} \\
\text{low} n_{\text{car}} & \text{high} & \text{high} \\
\text{high} n_{\text{car}} & \text{low} & \text{low}
\end{array} \]

\[ \begin{array}{c}
< \quad \approx \\
\approx \quad >
\end{array} \]

Rules \rightarrow lower efficiency, higher safety: sound!
GE-based rules generation

30 runs, each with 100 individuals, 100 generations

1 - ASR (Efficiency) vs $n_{\text{car}}$ (injected traffic)

CpT (Safety) vs $n_{\text{car}}$ (injected traffic)

Best GE rules:
- Slightly higher efficiency
- Higher safety
GE-based rules generation

30 runs, each with a 100 individuals, 100 generations

Best GE rules:
- slightly higher efficiency
- higher safety
Congestion

Experimental evaluation

Congestion: increasing injected traffic does not increase overall distance.

GE vs. hand-written at congestion: longer overall distance, less overall collisions.

Medvet et al. (UniTs) Road traffic rules with GE
Congestion

Overall collisions vs. Overall distance

One mark for each $n_{\text{car}}$ value
Congestion

One mark for each $n_{\text{car}}$ value

Congestion: increasing injected traffic does not increase overall distance
Congestion

One mark for each $n_{\text{car}}$ value

Congestion: increasing injected traffic does not increase overall distance

GE vs. hand-written at congestion:
- longer overall distance
- less overall collisions
Thanks!