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)

Driving

Human driver (*now*)

Driving

Human driver (*now*)

Driverless car (*the future*)

MACHINE LEARNING LAB

Driving

Human driver (*now*)

Driverless car (*the future*)

Both try to:

- get there
- avoid accidents
- comply to the rules

MACHINE LEARNING LAB

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?

MACHINE LEARNING LAB

Re-writing the 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?

Re-writing the 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*

MACHINE LEARNING LAB

In a nutshell

Evolutionary approach:

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

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

MACHINE LEARNING LAB

Table of Contents

The traffic model

Goals (and trade-offs):

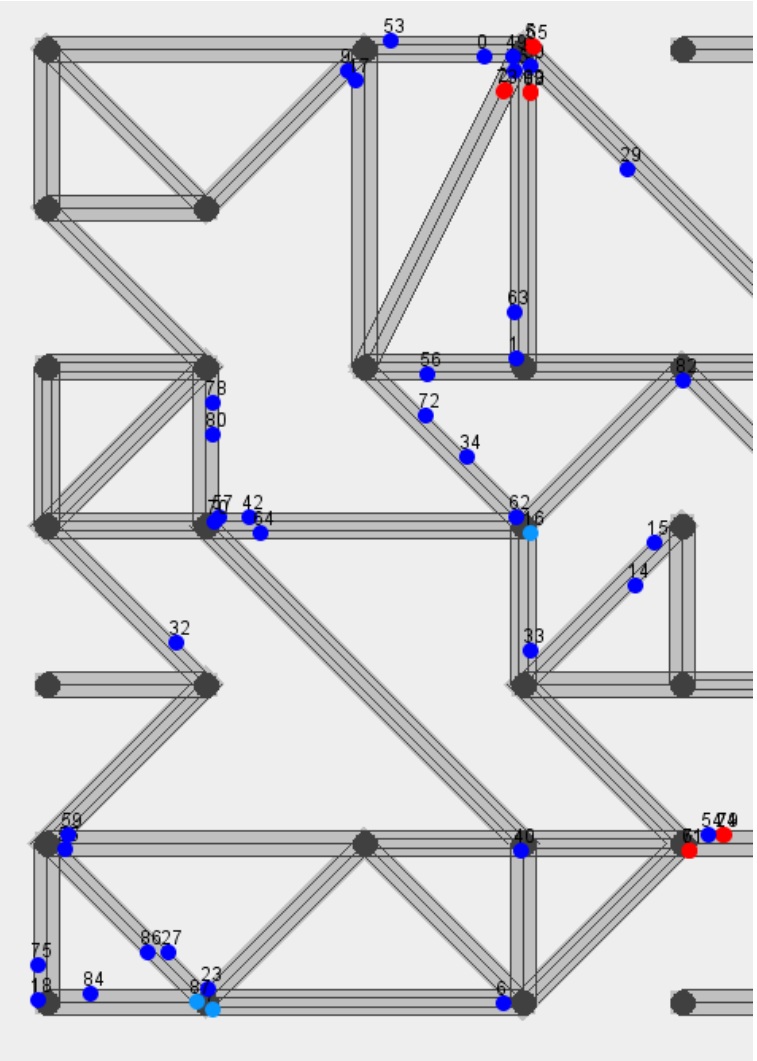
- detailed enough to include concepts amenable to be regulated
- simple enough to allow for simulation

Describes:

- the physical world
- the drivers' behavior

MACHINE LEARNING LAB

Physical world



Infrastructure:

- road sections and intersections
- lanes in sections

Cars:

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

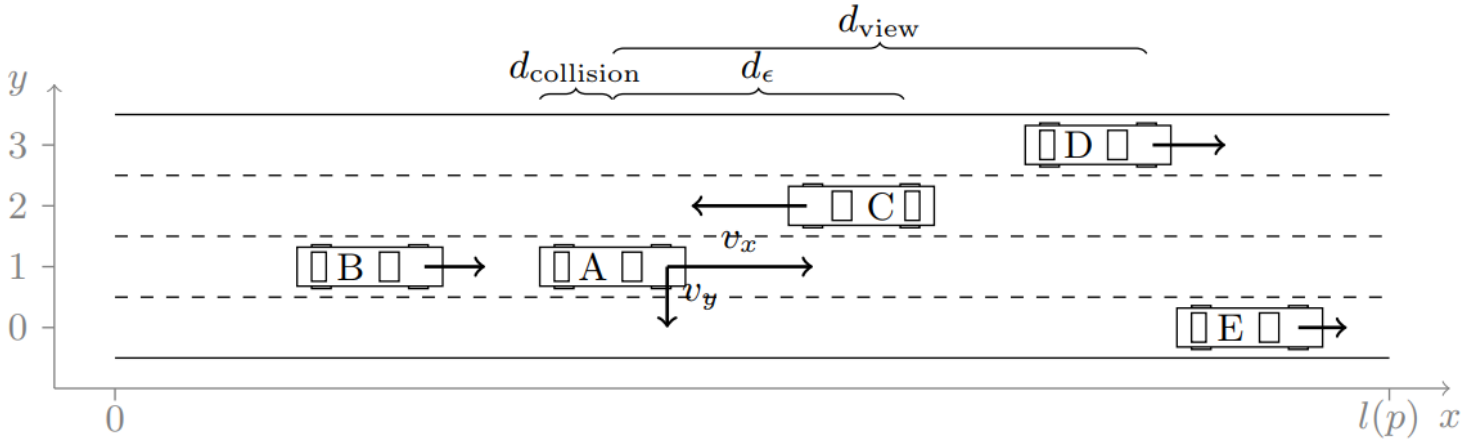
Collisions

With discrete time

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 ($\uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow, \leftarrow, \nwarrow$)



Drivers' behavior

Goal:

- just travel a target distance, no specific target position
- possibly at maximum (car) speed
- avoiding hitting other cars

MACHINE LEARNING LAB

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

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:

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

MACHINE LEARNING LAB

Language for the rules

As a context-free grammar:

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

Goal:

- express realistic rule like “stay on rightmost free lane”

MACHINE LEARNING LAB

Table of Contents

GE

We have:

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

$$\Downarrow$$

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: $f(R) = \alpha_{\text{time}}(1 - \text{ASR}) + \alpha_{\text{collision}}\text{CpT}$

MACHINE LEARNING LAB

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: $f(R) = \alpha_{\text{time}}(1 - \text{ASR}) + \alpha_{\text{collision}}\text{CpT}$

Computed over many (n_{sim}) simulations (stochasticity)

MACHINE LEARNING LAB

Table of Contents

Goals

Traffic model:

- is it sound?
- find values for parameters

MACHINE LEARNING LAB

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?

MACHINE LEARNING LAB

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?

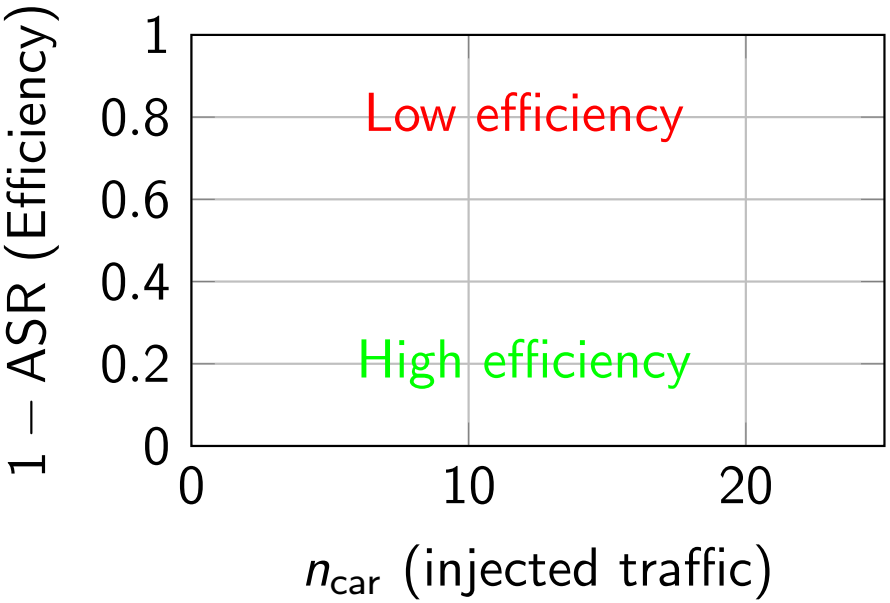
For both: how does injected traffic affect efficiency and safety?

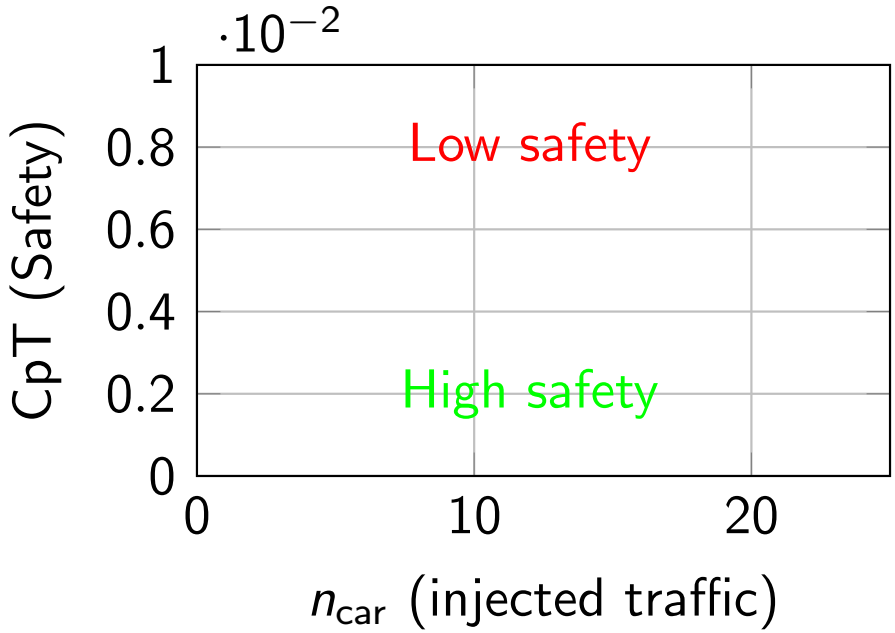
- congestion: no further increase in overall traveled distance

MACHINE LEARNING LAB

Traffic model validation

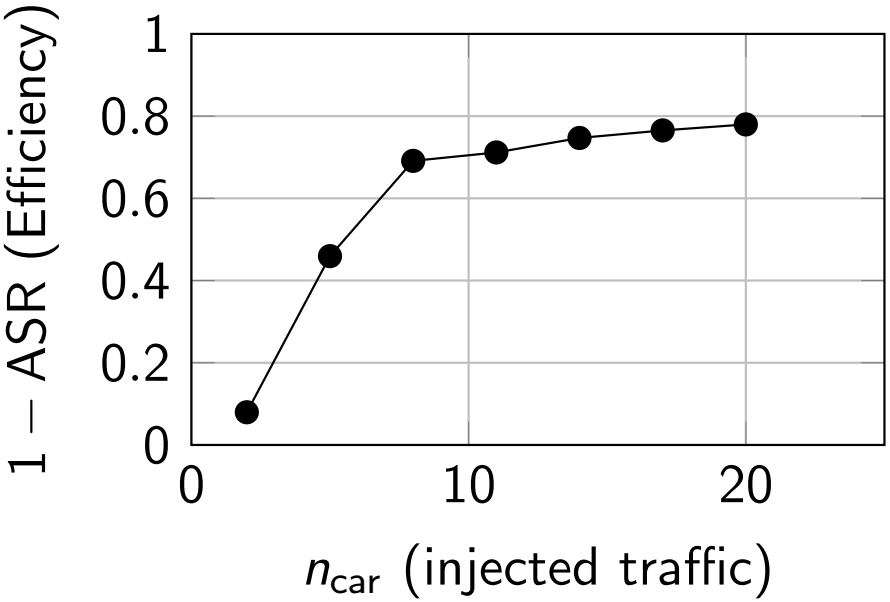
5 road sections, up to tens of cars, values averaged on 10 simulations

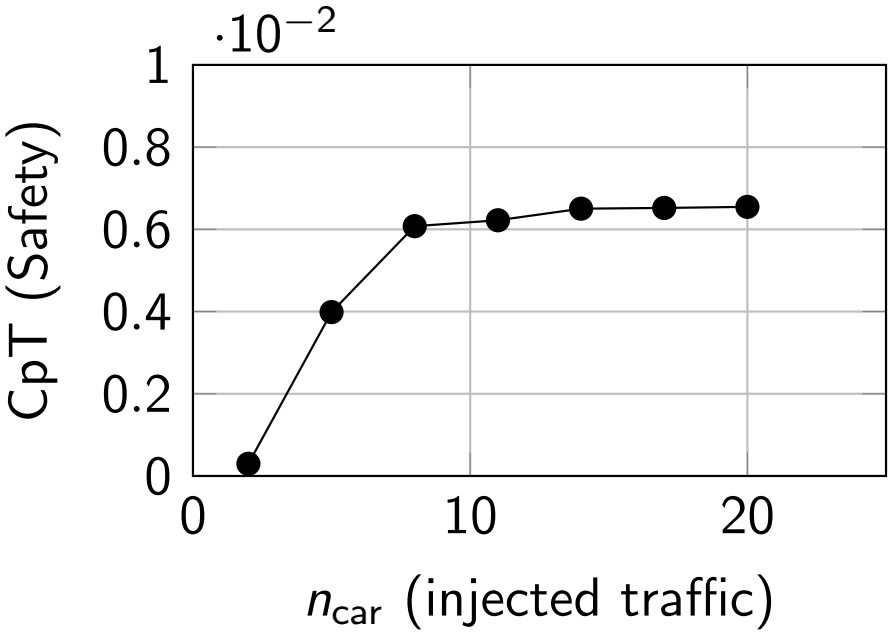




Traffic model validation

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



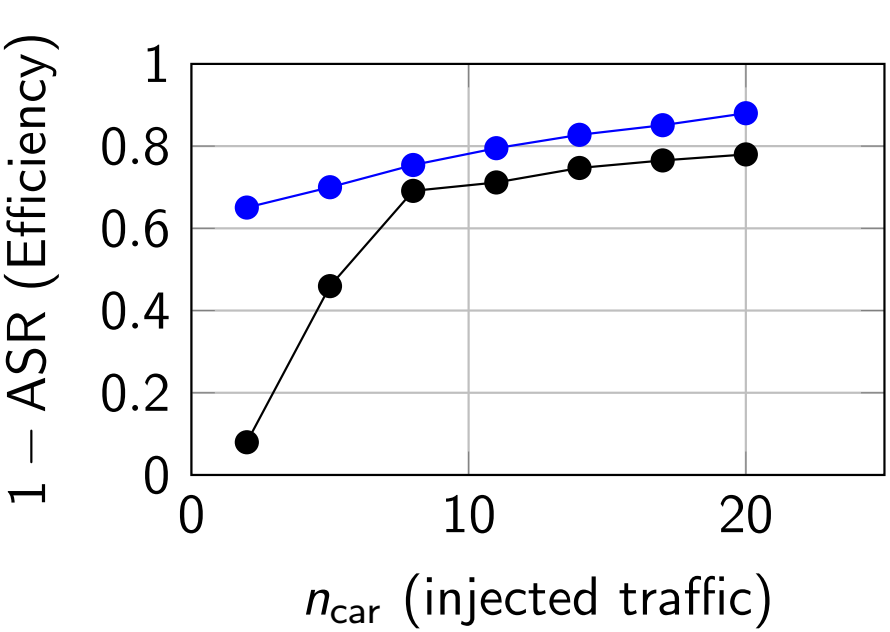


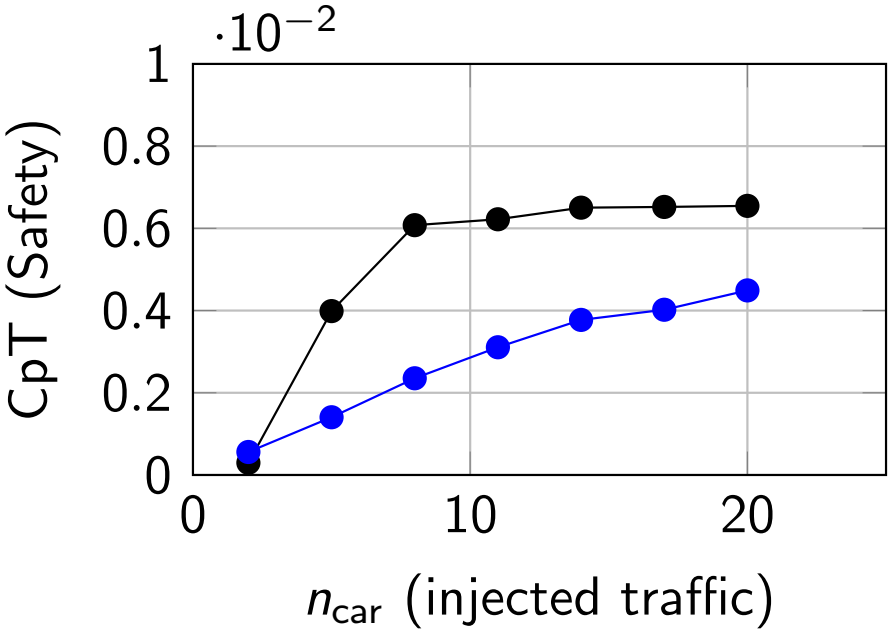
No rules

	Efficiency	Safety
low n_{car}	high	high
high n_{car}	low	low

Traffic model validation

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



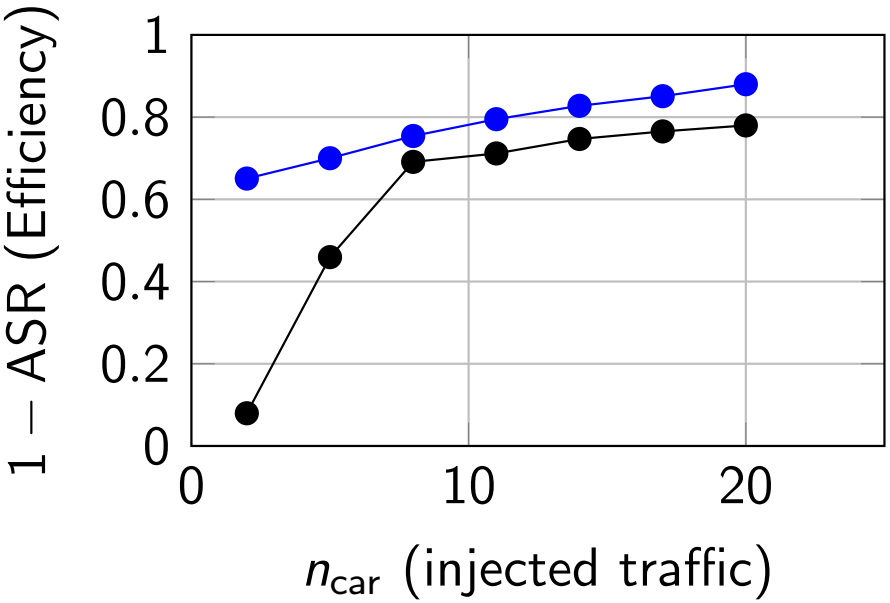


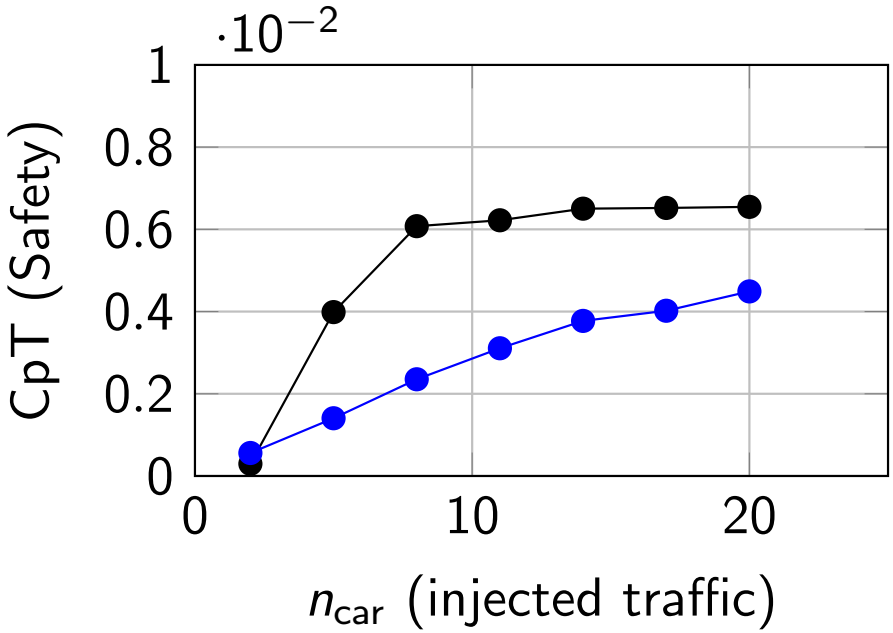
No rules — Hand-written rules

	Efficiency	Safety	Efficiency	Safety
low n_{car}	high	high	$<$	$\simeq$
high n_{car}	low	low	$\simeq$	$>$

Traffic model validation

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





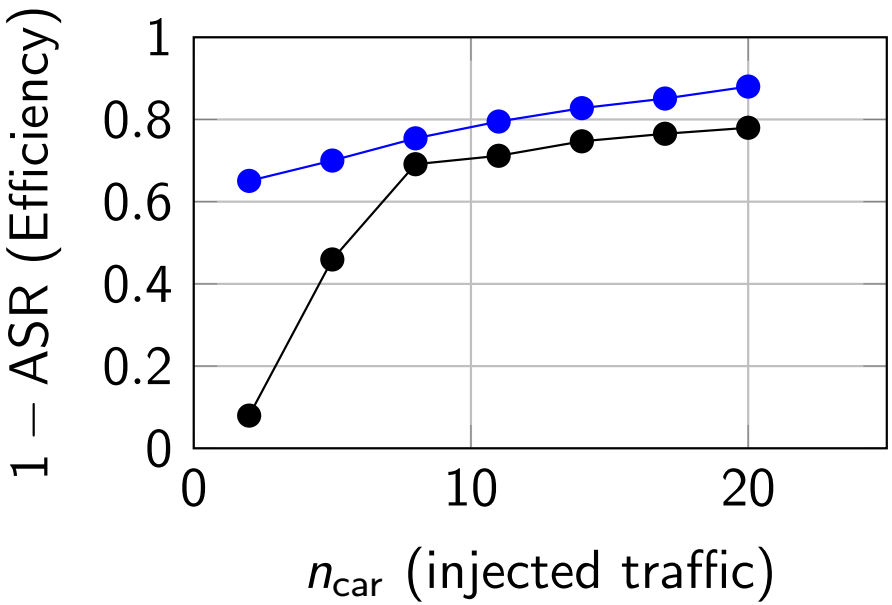
No rules — Hand-written rules

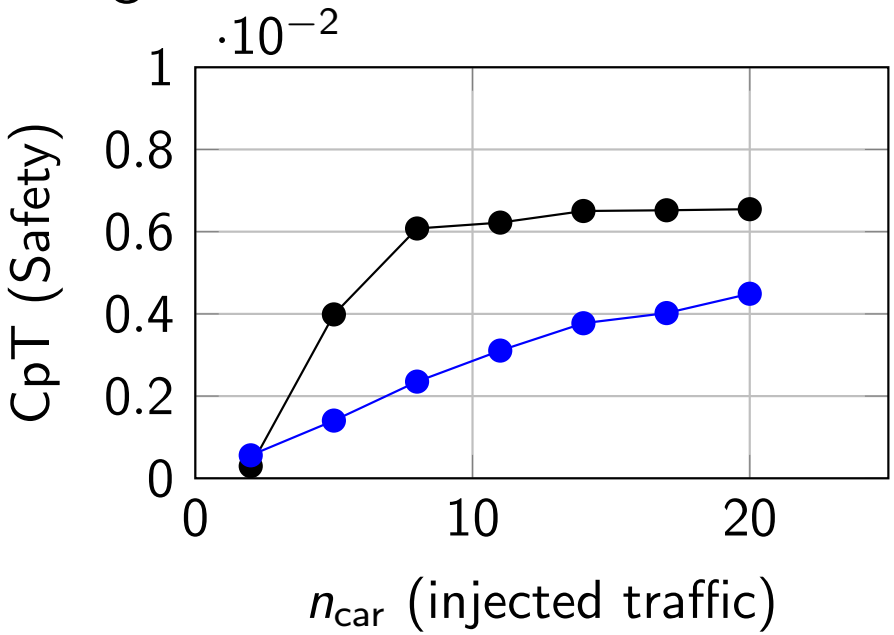
	Efficiency	Safety	Efficiency	Safety
low n_{car}	high	high	$<$	$\simeq$
high n_{car}	low	low	$\simeq$	$>$

Rules $\rightarrow$ lower efficiency, higher safety: sound!

GE-based rules generation

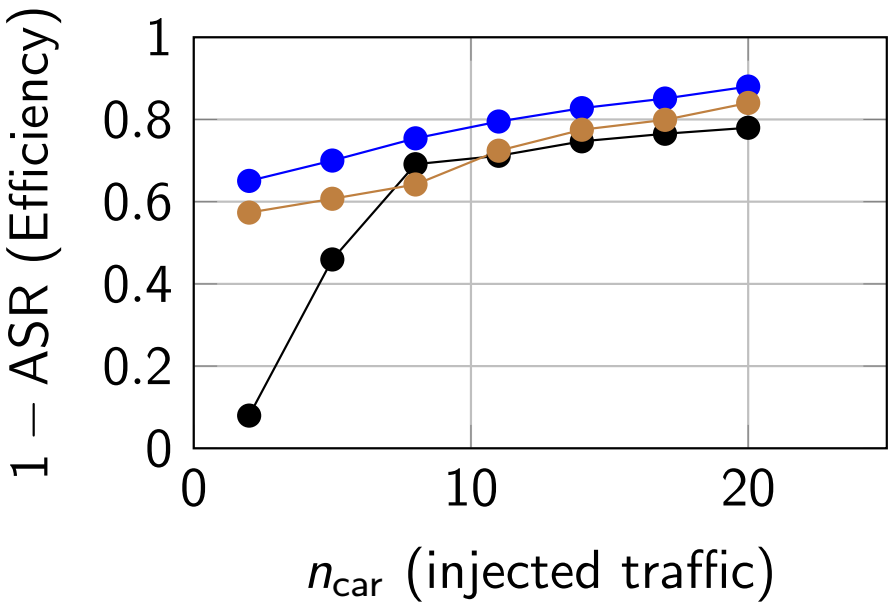
30 runs, each with a 100 individuals, 100 generations

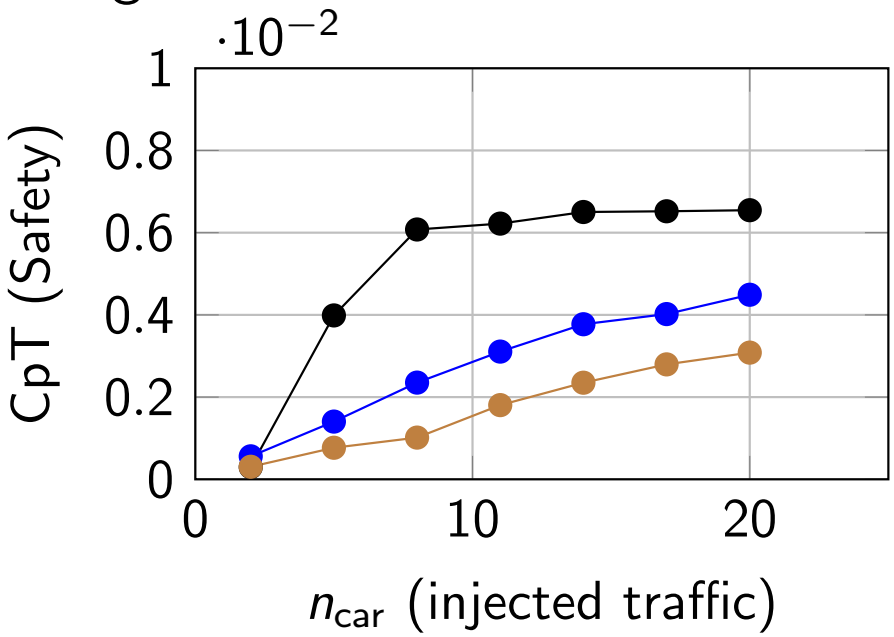




GE-based rules generation

30 runs, each with a 100 individuals, 100 generations

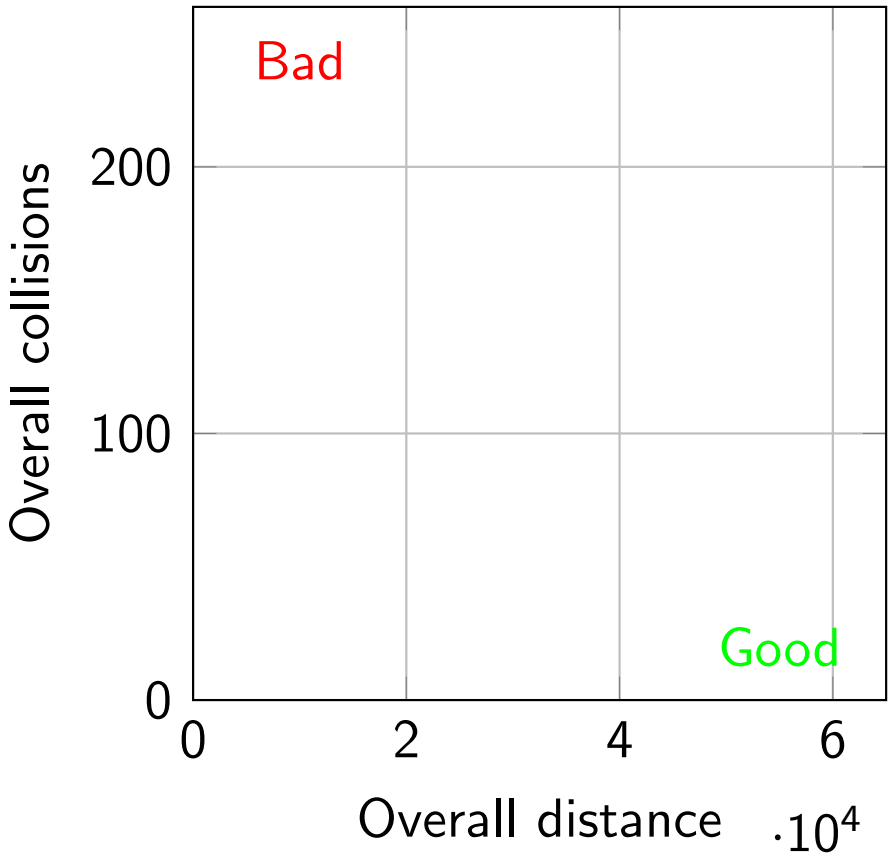




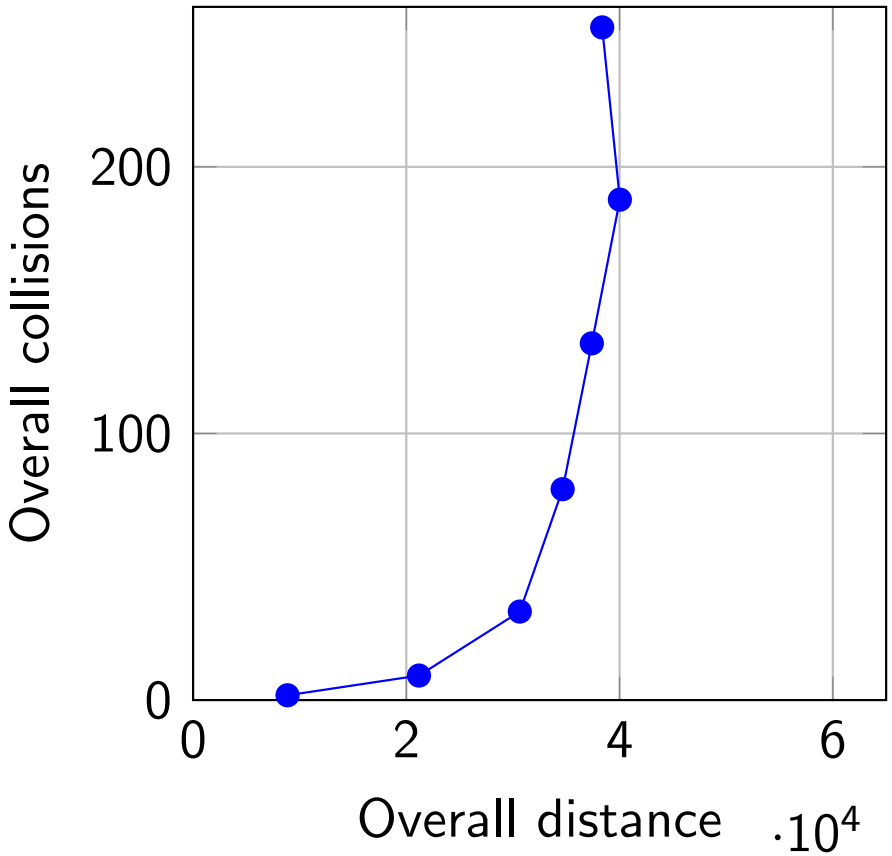
Best GE rules —●—:

- slightly higher efficiency
- higher safety

Congestion

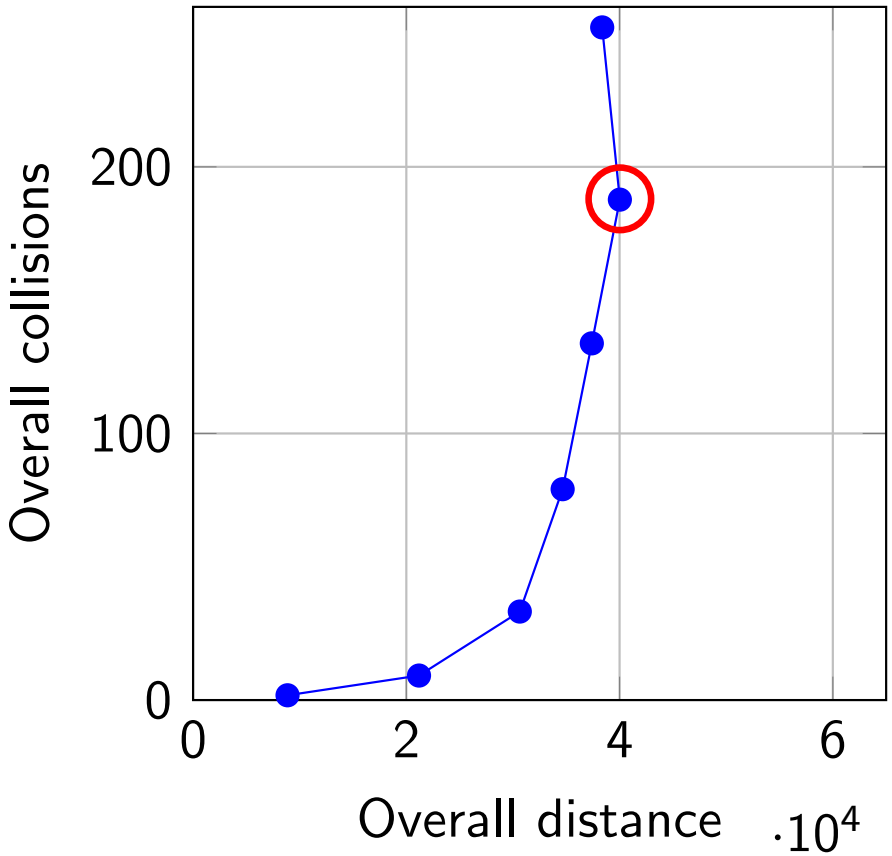

Bad
Good
Overall collisions
200
100
0
0
2
4
6
Overall distance
.10^4

Congestion



One mark for each n_{car} value

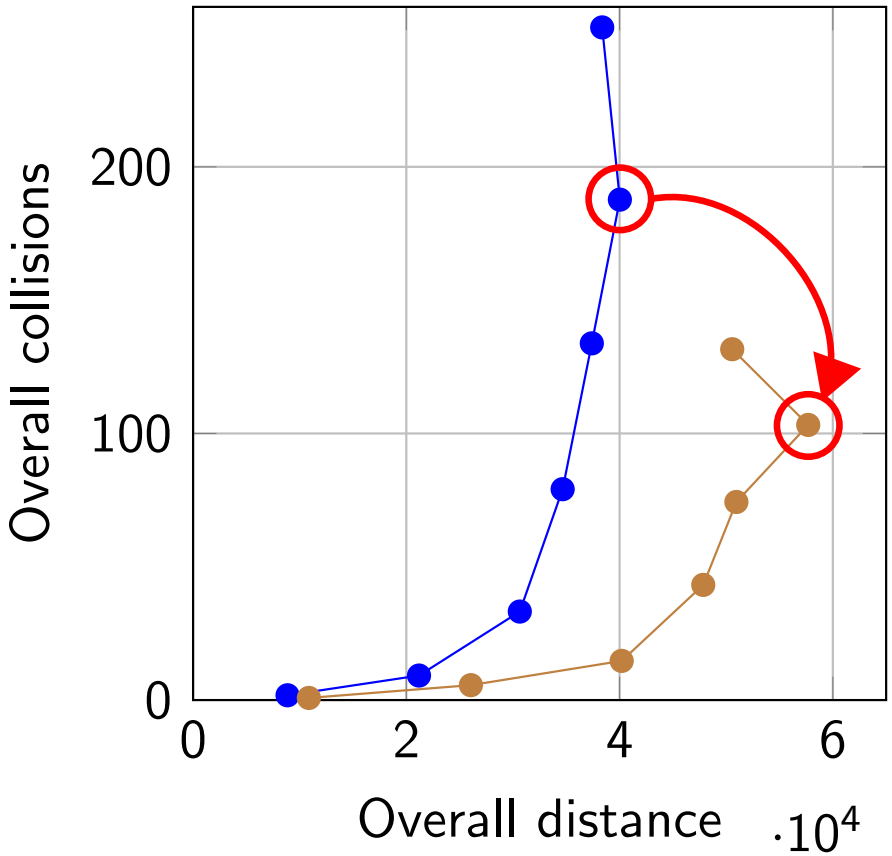
Congestion



One mark for each n_{car} value

Congestion: increasing injected traffic does not increase overall distance

Congestion



One mark for each n_{car} value

Congestion: increasing injected traffic does not increase overall distance

GE vs. hand-written at congestion:

- longer overall distance
- less overall collisions

 Thanks!