Selfish vs. Global Behavior Promotion in Car Controller Evolution

Jacopo Talamini, Giovanni Scaini, Eric Medvet, Alberto Bartoli

Department of Engineering and Architecture
University of Trieste
Italy

DTEO workshop @ GECCO, 16/7/2018, Kyoto (Japan)

http://machinelearning.inginf.units.it

Table of Contents

Cooperation and goals

Cooperative tasks:

- Achievement of the goal depends on the **cooperation** of many agents

Cooperation and goals

Cooperative tasks:

- Achievement of the goal depends on the **cooperation** of many agents

Learning agents controller:

- Neuroevolution: the controller is a **Neural Network**
- Its topology and weights are optimized by means of **Evolutionary Computation**

Cooperation and goals

Cooperative tasks:

- Achievement of the goal depends on the **cooperation** of many agents

Learning agents controller:

- Neuroevolution: the controller is a **Neural Network**
- Its topology and weights are optimized by means of **Evolutionary Computation**
  - there is a fitness function which drives the evolution

Cooperation and goals

Cooperative tasks:

- Achievement of the goal depends on the **cooperation** of many agents

Learning agents controller:

- Neuroevolution: the controller is a **Neural Network**
- Its topology and weights are optimized by means of **Evolutionary Computation**
  - there is a fitness function which drives the evolution

Should the fitness award selfish or global behavior?

Global vs. selfish behavior

What does the fitness assess?

Global behavior

Selfish behavior

Global vs. selfish behavior

What does the fitness assess?

Global behavior

+ captures the actual degree of achievement

Selfish behavior

Global vs. selfish behavior

What does the fitness assess?

Global behavior

+ captures the actual degree of achievement
- long to compute: simulate the full system with many agents

Selfish behavior

EVOLUTIONARY ROBOTICS ARTIFICIAL LIFE LAB

Global vs. selfish behavior

What does the fitness assess?

Global behavior

+ captures the actual degree of achievement
- long to compute: simulate the full system with many agents

Selfish behavior

+ "fast" to compute: simulate just one agent

Global vs. selfish behavior

What does the fitness assess?

Global behavior

+ captures the actual degree of achievement
- long to compute: simulate the full system with many agents

Selfish behavior

+ "fast" to compute: simulate just one agent
- a proxy for the actual degree of achievement: how to choose?

An example

Road traffic system:

- Agents: car drivers
- System goal (two-fold):
  - reaching targets (efficiency)
  - avoiding collisions (safety)

An example

Road traffic system:

- Agents: car drivers
- System goal (two-fold):
  - reaching targets (efficiency)
  - avoiding collisions (safety)

Implicitly cooperative

An example

Road traffic system:

- Agents: car drivers
- System goal (two-fold):
  - reaching targets (efficiency)
  - avoiding collisions (safety)

Implicitly cooperative

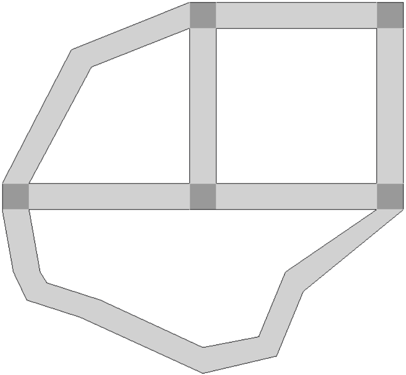
Global or selfish fitness while evolving a cooperative driver?

Table of Contents

EVOLUTIONARY ROBOTICS ARTIFICIAL LIFE LAB

World

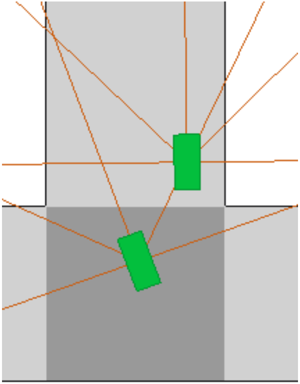
- 2D continuous space, discrete time world
- road (section, intersection), off-road
    - cars move on road only
    - cars collide with other cars and with road side



Car and agents

Car:

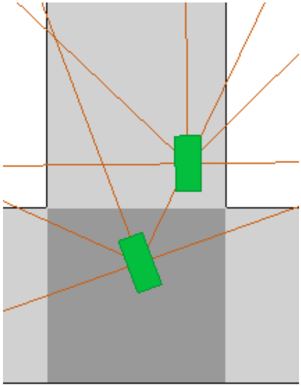
- each w/ 5×3 distance sensors
  - to roadside
  - to intersection
  - to other cars



Car and agents

Car:

- each w/ 5×3 distance sensors
  - to roadside
  - to intersection
  - to other cars



Agent (car driver/controller):

- assigned to sequence of targets
  - subsequent targets on adjacent road sections

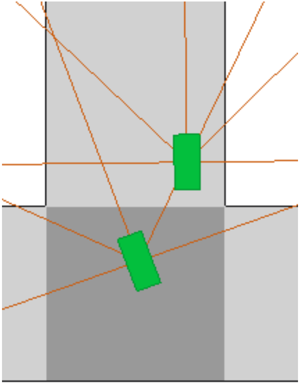
Car and agents

Car:

- each w/ 5×3 distance sensors
  - to roadside
  - to intersection
  - to other cars

Agent (car driver/controller):

- assigned to sequence of targets
  - subsequent targets on adjacent road sections
- neural network
  - input: 5×3 distance sensors, car speed, distance to target, direction of target
  - output: steering angle, acceleration/brake



EVOLUTIONARY ROBOTICS ARTIFICIAL LIFE LAB

Global goal

Maximize traffic efficiency and safety

Global goal

Maximize traffic efficiency and safety

- Efficiency: the global number of reached targets in the unit of time

$$E = \frac{1}{n_{\text{car}}} \sum_{j=1}^{n_{\text{car}}} \frac{1}{\tau} \left(t_j + 1 - \frac{l_j^f}{l_j^i} \right)$$

Global goal

Maximize traffic efficiency and safety

- Efficiency: the global number of reached targets in the unit of time

$$E = \frac{1}{n_{\text{car}}} \sum_{j=1}^{n_{\text{car}}} \frac{1}{\tau} \left(t_j + 1 - \frac{l_j^f}{l_j^i} \right)$$

- Safety: (opposite of) the global number of collisions in the unit of time

$$S = -\frac{1}{n_{\text{car}}} \sum_{j=1}^{n_{\text{car}}} \frac{c_j}{\tau}$$

Neuroevolution of the agents

Goal: evolve a the driver which maximizes, on average, traffic efficiency and safety

We used NEAT

Neuroevolution: global

- individual: controller
- fitness:
  - $f_{\text{glob}} = 100E + 0.1S$
  - n_{sim} simulations with n_{car} each, **all with the same controller** under evaluation

Neuroevolution: selfish

- individual: controller
- fitness:
  - $f_{\text{self}} = 100 E_{\text{self}} + 0.1 S_{\text{self}}$

$$E_{\text{self}} = \frac{1}{\tau}\left(t + 1 - \frac{l^f}{l^i}\right) \qquad S_{\text{self}} = -\frac{c}{\tau}$$

  - the controller under evaluation inserted in n_{sim} simulations with other $n_{\text{car}} - 1$ **different controllers**

EVOLUTIONARY ROBOTICS ARTIFICIAL LIFE LAB

Global vs. selfish

Global

- one simulation to assess one controller

Selfish

- one simulation to assess n_{car} controllers at once, but on a proxy of their actual goal

Table of Contents

Two stages

Evolution:

- $n_{\text{sim}} = 3$, $n_{\text{car}} = 20$, $\tau = 30\,\text{s}$ (simulated)
- 10 runs, same wall time (24 h) per run for both approaches

Validation of best evolved controllers $(10 + 10)$:

- $n_{\text{sim}} = 10$, $n_{\text{car}} = \{5, \dots, 50\}$, $\tau = 60\,\text{s}$ (simulated)

Two stages

Evolution:

- $n_{\mathrm{sim}} = 3$, $n_{\mathrm{car}} = 20$, $\tau = 30\,\mathrm{s}$ (simulated)
- 10 runs, same wall time (24 h) per run for both approaches

Validation of best evolved controllers $(10 + 10)$:

- $n_{\mathrm{sim}} = 10$, $n_{\mathrm{car}} = \{5, \dots, 50\}$, $\tau = 60\,\mathrm{s}$ (simulated)
  - range of $n_{\mathrm{car}} \rightarrow$ "robustness" of evolved controllers to traffic conditions

Two stages

Evolution:

- $n_{\text{sim}} = 3$, $n_{\text{car}} = 20$, $\tau = 30\,\text{s}$ (simulated)
- 10 runs, same wall time (24 h) per run for both approaches

Validation of best evolved controllers $(10 + 10)$:

- $n_{\text{sim}} = 10$, $n_{\text{car}} = \{5, \dots, 50\}$, $\tau = 60\,\text{s}$ (simulated)
  - range of $n_{\text{car}} \rightarrow$ "robustness" of evolved controllers to traffic conditions
- homogeneous vs. heterogeneous
  - homogeneous: all cars driven by the same evolved controller
  - heterogeneous: 50% of cars driven by random (unskilled) controllers

Two stages

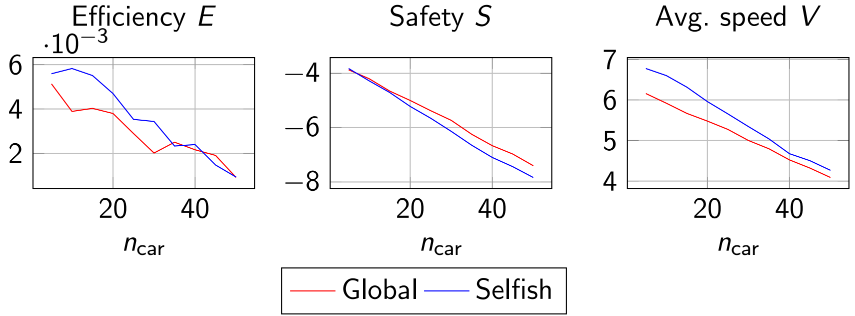
Evolution:

- $n_{\text{sim}} = 3$, $n_{\text{car}} = 20$, $\tau = 30\,\text{s}$ (simulated)
- 10 runs, same wall time (24 h) per run for both approaches

Validation of best evolved controllers $(10 + 10)$:

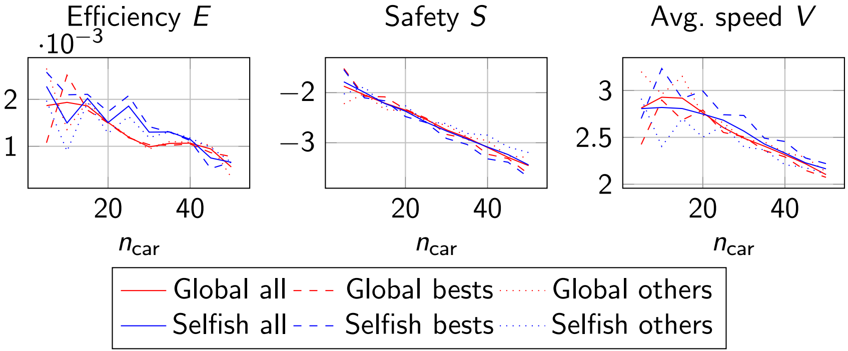
- $n_{\text{sim}} = 10$, $n_{\text{car}} = \{5, \dots, 50\}$, $\tau = 60\,\text{s}$ (simulated)
  - range of $n_{\text{car}} \rightarrow$ "robustness" of evolved controllers to traffic conditions
- homogeneous vs. heterogeneous
  - homogeneous: all cars driven by the same evolved controller
  - heterogeneous: 50% of cars driven by random (unskilled) controllers
  - "robustness" of evolved controllers to other driving behaviors

Homogeneous validation



- In general: the heavier the traffic, the lower efficiency and safety
- Trade-off between efficiency and safety: selfish are more efficient, less safe (reasonable)
- Selfish better with light traffic: more capable of "driving alone"
- Selfish always drive faster

Heterogeneous validation



- Fuzzier difference
- With medium traffic: selfish more efficient, equally safe → robust to presence of unskilled drivers?

Conclusions

In cooperative tasks tackled with neuroevolution:

- selfish-based fitness may replace global-based fitness
- opportunity for robustness
- how to choose a proper selfish-based fitness?

Thanks!