Selfish vs. Global Behavior Promotion in Car Controller Evolution

Jacopo Talamini
DIA - Università di Trieste
Trieste, Italy
talamini.jacopo@gmail.com

Giovanni Scaini
DIA - Università di Trieste
Trieste, Italy
giovanni.scaini@gmail.com

Eric Medvet
DIA - Università di Trieste
Trieste, Italy
emedvet@units.it

Alberto Bartoli
DIA - Università di Trieste
Trieste, Italy
bartoli.alberto@units.it

ABSTRACT

We consider collective tasks to be solved by simple agents synthesized automatically by means of neuroevolution. We investigate whether driving neuroevolution by promoting a form of selfish behavior, i.e., by optimizing a fitness index that synthesizes the behavior of each agent independent of any other agent, may also result in optimizing global, system-wide properties. We focus on a specific and challenging task, i.e., evolutionary synthesis of agent as car controller for a road traffic scenario. Based on an extensive simulation-based analysis, our results indicate that even by optimizing the behavior of each single agent, the resulting system-wide performance is comparable to the performance resulting from optimizing the behavior of the system as a whole. Furthermore, agents evolved with a fitness promoting selfish behavior appear to lead to a system that is globally more robust with respect to the presence of unskilled agents.

CCS CONCEPTS

• **Computing methodologies → Cooperation and coordination**; *Multi-agent systems*; *Neural networks*; • **Computer systems organization →** *Robotic autonomy*;

KEYWORDS

NEAT, Neuroevolution, Driverless car, Collective behavior

ACM Reference Format:
Jacopo Talamini, Giovanni Scaini, Eric Medvet, and Alberto Bartoli. 2018. Selfish vs. Global Behavior Promotion in Car Controller Evolution. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3205651.3208254

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208254

1 INTRODUCTION

Recent developments in robotics are increasingly focusing on tasks that should be solved by means of *cooperation* among a set of autonomous, simple robots [2, 10]. These collective robotics frameworks are often based on *neuroevolution* [9], that is, on robotics controllers based on artificial neural networks, whose structure is optimized by means of evolutionary procedures. Neuro Evolution of Augmenting Topologies (NEAT) [12] is one of the most popular approaches in this respect, which evolves neural network-based controllers by tuning their weights and, to some extent, their topology, based on the task-specific *fitness* exhibited by controllers. Such indexes can be defined based on the global behavior of the system, thus their computation is complex and expensive.

In this work, we investigate the feasibility of driving the evolutionary search by promoting selfish behavior, hence decomposing the problem of collective evaluation into fitness indexes based on the behavior of single individuals. In other words, we intend to assess experimentally whether promotion of forms of selfish behaviors, i.e., optimization of *local* properties, may also result in optimizing *global*, system-wide properties. We focus our analysis on a specific and challenging task, i.e., evolutionary synthesis of a car controller for a road traffic scenario. We simulate a road graph crossed by a number of cars at the same time, where each car is given a sequence of *targets*, i.e., positions in the graph, that have to be reached. Each car is equipped with sensors describing the current kinematic properties of the car, the distance from the closest car, from the roadside, from the next target, and alike. The behavior of a car is controlled by a neural network that takes the sensors readings as input and emits outputs that determine acceleration and steering angle.

We assess experimentally two radically different alternatives for driving the neuroevolution, one in which the fitness is defined based on the behavior of each car (*agent*) independent on the behavior of all other agents, the other in which the fitness quantifies the global behavior of *all* the agents. The computation of the former may exploit a much greater degree of parallelism than the latter, making it an attractive option in practice. Our results indicate that the two approaches lead to similar global performance, that is, although the task performance has to be computed system-wide, the performance resulting from optimizing the behavior of each single agent is similar to the one resulting from optimizing the behavior of the system as a whole. Furthermore, agents evolved with a locally-defined fitness appear to lead to a system that is globally more robust in the sense that the system-wide performance decrease induced by increasing amounts of unskilled agents is smaller.

2 RELATED WORKS

To the best of our knowledge, there are not other studies facing the problem of fitness decomposition in a collective environment in the way we present here. However, there are several previous works which are relevant to the present study w.r.t. at least one of the following three research topics: (a) global vs. selfish behavior, (b) neuroevolution, (c) and autonomous car controllers (a.k.a. driverless cars). We here briefly survey some recent works concerning each of those topics.

Collective behavior is an aspect that appears in many studies related to animals and human groups. This kind of behavior has been studied by Lukeman et al. [6] in particular considering aggregation patterns and collective intelligence. Benefits of aggregation as a form of collective behavior have been studied by Parrish et al. [11] with focus on organization and decision-making patterns inspired by complexity theory. Another computational model for collective behavior has been presented by Goldstone et al. [4] as a promising way to give information on how individuals decisions lead to organization into groups. Agent-based modelling of simulated system with collective behavior has been deeply discussed by Bonabeau [1], in particular regarding real-world applications.

Concerning neuroevolution, NEAT is one of the most widespread approaches. It has been presented by Stanley and Miikkulainen [12] as a method for the optimization of neural networks capable of outperforming preexisting fixed-topology networks on reinforcement learning tasks. A method for co-evolution of cooperative agents called Enforced Subpopulation (ESP) has been presented by Yong and Miikkulainen [15]. They consider a scenario in which many agents have to coordinate their behavior to achieve a common goal. The solution proposed by ESP is based on separating agents into sub-populations and later testing them together in the same task. A different approach is to use a collective-based neuroevolution like in the work by Nitschke et al. [8], where the authors present the results of a collective neuroevolution controller design method applied to a multi-rover task. In the cited work, the authors focus on collective behavior tasks and on behavior specialization between controllers. Both these two methods have been evaluated by Van Krevelen et al. [14] in a collective construction task, proving effectiveness of neuroevolution in such collective environments. Another evolutionary approach for artificial neural networks has been developed by Thangavelautham et al. [13] for robotic tasks. The authors give the networks a global fitness function only: they show that bigger modular Emergent Task Decomposition strategies Networks (ETDN) outperform smaller networks not based on these strategies on collective tasks. ETDN is able to decompose a complex task into simpler ones through competition and self-organization.

Finally, driverless cars has been a central research topic tackled from many points of view. Social and ethical aspects of autonomous vehicles have been studied by Holstein et al. [5] by dealing with the decision making process. A rather different approach for pursuing global efficiency and safety has been proposed by Medvet et al. [7]: instead of learning a controller, the authors of the cited paper propose to learn the traffic rules which could, if applied to existing controllers, lead to better efficiency and safety. Figueiredo et al. [3] present an approach for implementing a simulator for driverless cars; moreover, they include a study of the state of the art of driverless cars in simulations, focusing on some aspects as efficiency and collision avoidance.

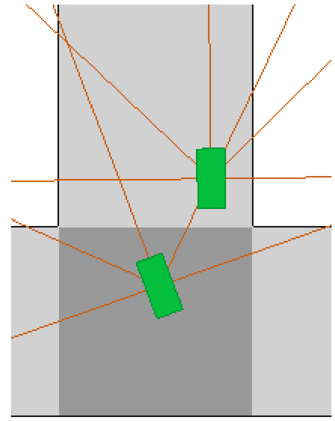

Figure 1: Representation of two cars crossing a road intersection (darker area), equipped both with 5 radial proximity sensors. The sensors detect the distance to the roadsides, intersections, and other cars.

3 CAR CONTROLLER EVOLUTION

3.1 Traffic model

We consider a scenario with continuous space and discrete time in which a number of cars move along the roads, possibly colliding with other cars or roadsides. Cars move in a finite and two-dimensional space, partitioned in regions. Each region can be either a *road* or *off-road*. Cars can never be in an off-road region and can freely move in road regions. A road region can be of type *road section* or *road intersection*. A *roadside* divides a road region from an adjacent off-road region. Cars cannot cross roadsides.

A *car* is a rigid body that can move along the roads and collide with other cars or roadsides. A car has geometrical features (e.g., shape) and physical features (e.g., mass, density, friction, and restitution). At each time step a car is defined by the position of its center of mass, its orientation, its linear velocity and its steering angle.

Each car is equipped with radial proximity *sensors* equally spaced in an angle of 180° (see Figure 1). Each sensor, defined by an angle relative to the orientation of the car and a range, detects three distances: (a) from the closest car, (b) from the closest roadside, and (c) from the closest intersection. Sensors have a maximum range. In case an object is not within the sensor range, the detected distance is equal to the sensor range. The sensors are ideal, i.e., they are not affected by noise. Each car has a *controller* that takes the distances detected by sensors as input and whose behavior is described below.

Each car is assigned a sequence of *targets*. A target is a line on a road section that the car has to cross. Once the car reaches its target, the car is immediately assigned a new target. The new target is located in a road section close to the one where the car is currently located and selected at random, in the same direction the car is moving.

At each time step the position of the car is updated according to its steering angle, its linear velocity at the previous time step, and the time elapsed from it. A *collision* occurs when two or more car shapes overlap or a car shape intersects a roadside. Every car involved in the collision is penalized by a reduction of the linear speed.

The car *controller* is an algorithm that determines the steering and acceleration/braking of the car. In particular, at each time step, the controller processes the following input data: distances from other cars, roadsides, and intersections as detected by the sensors; linear speed; target direction, i.e., difference between current orientation of the car and angle to the target; and distance from the target. Based on this data, the controller produces the output that is interpreted as driving commands for the car: steering angle and acceleration (possibly negative, i.e., braking) to be applied to the car.

In this work, we consider a controller based on an artificial neural network (ANN), in which the input and output layers are predefined. In particular, the ANN-based controller has three input neurons for each car sensor (distance from other cars, roadsides, intersections) and three more input neurons: current car speed, distance from the target, and target direction. The controller has two output neurons: one controls the steering angle, the other controls acceleration of the car.

3.2 Traffic evaluation

We are interested in the driving task as a collective task. That is, we consider the set of cars as a mean for transporting goods or persons towards their respective targets by moving along the roads. Hence, one can measure how well this collective task is performed by considering the traffic *efficiency*, i.e., the average number of reached targets. On the other hand, it is also important to consider the traffic *safety*, i.e., the number of collisions among cars and with roadsides.

Based on these considerations, we quantify efficiency and safety as follows. Given the road graph and a set of n_{car} car controllers, we perform a number of n_{sim} simulations, each lasting τ simulated seconds, and we measure, for each j-th car: the number c_j of collisions in which the car was involved, the overall number t_j of targets reached, the initial and final distances l_j^i and l_j^f between the car and the last, not reached target when the simulation ended. Then, we define efficiency as:

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

and we define safety as:

$$S = -\frac{1}{n_{\text{car}}} \sum_{j=1}^{n_{\text{car}}} \frac{c_j}{\tau} \quad (2)$$

The indexes are averaged across the n_{sim} simulations. For both of them, the greater, the better: note that S is, by design, negative, whereas E may assume negative values if the car moves away, rather than towards, the last target.

We emphasize that the chosen indexes quantify the performance of the system as a whole and that maximizing the performance of a single car may result in lower collective performance. Intuitively, a car controller which is able to drive fast, and which can hence reach many targets, might be a danger for the other cars, eventually making their collective in-efficiency larger than its individual high efficiency.

3.3 Global and selfish neuroevolution

We evolve car controllers by means of NEAT [12], one of the most widespread neuroevolution approaches, which simultaneously evolves both the topology and the weights of a neural network. NEAT includes a principled method of crossover of different network topologies, based on the assignment of incremental innovation numbers, protection of structural innovations through speciation, and incremental growing from minimal structure. Initially, networks generated by NEAT have inputs directly connected to outputs, with a specified and fixed number of inputs and outputs. During the evolution, these networks are subjected to mutation and crossover in their topology as well as in their parameters—i.e., NEAT can perturb the weight of a synapse, can add or remove neurons, or even connections between neurons.

NEAT is a single-objective optimization method, thus the neuroevolution is driven by a single fitness index. By *global* neuroevolution we indicate the scenario in which the fitness index quantifies the aggregate behavior of all car controllers (i.e., networks evolved by NEAT), while by *selfish* neuroevolution we indicate the scenario in which the fitness index quantifies the behavior of a single car controller independent of the behavior of the other controllers.

In detail, we define the fitness for the global neuroevolution as:

$$f_{\text{coll}} = 100E + 0.1S \quad (3)$$

where 100 and 0.1 are two coefficient arbitrarily chosen based on some preliminary experimentation and with the aim of providing similar weighted values for the two indexes. We define the fitness for the selfish neuroevolution as:

$$f_{\text{self}} = 100E_{\text{self}} + 0.1S_{\text{self}} \quad (4)$$

where:

$$E_{\text{self}} = \frac{1}{\tau} \left(t + 1 - \frac{l^f}{l^i} \right) \quad (5)$$

$$S_{\text{self}} = -\frac{c}{\tau} \quad (6)$$

Values for t, l^i, l^f, and c are obtained by observing the behavior of a car moved by the controller to be evaluated when inserted in a simulation with $n_{\text{car}} - 1$ other cars (possibly controlled by different controllers, obtained from the individuals of the same population) lasting τ simulated seconds. As for E and S, we compute E_{self} and S_{self} by averaging the observations over n_{sim} simulations.

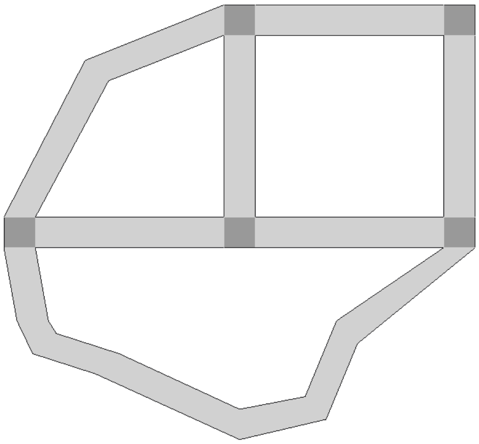

Figure 2: The road track used in the experimentation: the intersections are represented in dark gray, the sections in light gray, the roadside in black. The track bounding box measures approximately 200 m × 200 m.

Table 1: Parameters of the traffic model.

Parameter	Value
Mass of the car	1500 Kg
Size of the car	3.8 m × 1.7 m
Number of sensors per car	5
Range of the sensors	50 m
Maximum speed of the car	$42\ \frac{\text{m}}{\text{s}}$
Speed reduction of the collided car	90%
Maximum acceleration of the car	$6\ \frac{\text{m}}{\text{s}^2}$
Maximum steering angle per time step	$\frac{\pi}{36}$
Time step of the simulation	0.05 s

4 EXPERIMENTAL EVALUATION

In order to perform the experimental analysis, we developed a traffic simulator in Java which implements the traffic model presented in Section 3.1 and is internally based on the dyn4j Java two-dimensional physics engine[1]. We performed all the experiments using a road graph with 5 intersections enclosed in a bounding box of approximately 200 m × 200 m (see Figure 2) and with the parameters shown in Table 1.

We used the standard Java implementation of NEAT[2] for performing the neuroevolution with a population of 100 individuals. For both the evolution driven by the global fitness f_{coll} and the one driven by the selfish fitness f_{self}, we gave NEAT a time budget of 24 h (wall time) after which we stopped the evolution. We performed the experiments on the CINECA HPC cluster Marconi-A1, one node for each evolutionary run, the node having 2 Intel Xeon E5-2694 v4 CPUs (2.3 GHz) with 18 cores each and 128 GB of RAM. We recall that the computation of a single fitness value takes much longer with f_{coll} than with f_{self}: we observed that NEAT performed, in the given time budget, ≈ 80 generations with the former and ≈ 1800 generations with the latter. When computing the fitness, we set $n_{\text{sim}} = 3$, $n_{\text{car}} = 20$, $\tau = 30$ s (simulated).

We performed 10 evolutionary runs for each of the two variants (global and selfish). At the end of each of the 20 runs, we took the best resulting controller (i.e., the one with the greatest fitness value) and assessed it as follows. We performed 10 simulations with longer simulation time than the evolution ($\tau = 60$ s instead of $\tau = 30$ s) on the same track and with a number n_{car} of cars ranging from 5 to 50 (with steps of 5) and in two scenarios. In the *homogeneous* scenario, all the n_{car} cars in the simulation were controlled by the same ANN learned by NEAT, i.e., the controller being assessed. In the *heterogeneous* scenario, half of the n_{car} cars in the road graph were controlled by the controller being assessed while the remaining half were controlled by $\frac{\text{car}}{2}n$ different controllers; we randomly selected these not skilled controllers among the individuals from the early generations of the runs performed. After each simulation of the validation, we measured the efficiency and safety of the observed traffic, as described by Equations 1 and 2, as well as the average speed, measured as:

$$V = \frac{1}{n_{\text{car}}} \sum_{j=1}^{n_{\text{car}}} \frac{l_j}{\tau} \tag{7}$$

where l_j is the overall travelled distance by the j-th car.

The rationale for the validation is two-fold. On one hand, we were interested in verifying to which degree the goal was achieved of building a controller able to maximize the traffic efficiency and safety. On the other hand, we wanted to investigate how robust the generated controller was w.r.t. (a) different levels of injected traffic (ranging from light with $n_{\text{car}} = 5$ to heavy with $n_{\text{car}} = 50$) and (b) the presence in the road track of other cars controlled by different, worse controllers.

4.1 Results and discussion

4.1.1 Evolution. Figure 3 summarizes how the neuroevolution proceeded in the two cases, by showing the fitness value of the best individual in the population vs. the elapsed time from the beginning of the evolution (fitness values are averaged across the 10 evolutionary runs).

It can be seen that the actual values of the fitness are clearly larger in the selfish variant. This result is consistent with the way the fitness is computed: in the selfish variant, f_{self} is based on the efficiency and safety of a single car which, in fortunate conditions, may score well. In contrast, in the global variant, f_{coll} takes into account the efficiency and safety of the whole traffic, hence averaging fortunate and unfortunate conditions which occur for the different cars. As a further evidence, it can be seen that the starting values in the two cases are different, f_{self} being $\approx 10 f_{\text{coll}}$. From another point of view and beyond the actual values, this finding suggests that the selfish variant might overestimate a controller driving ability because of the fortunate conditions, making f_{self} less capable of driving the evolution.

Figure 3 also shows that the improvement of the fitness in the global variant appears to proceed more slowly. Indeed, due to the long selfish evaluation time with f_{coll}, less than one hundred generations are evolved, whereas several hundreds are evolved with f_{self}.

[1] http://www.dyn4j.org/
[2] https://github.com/encog/encog-java-core

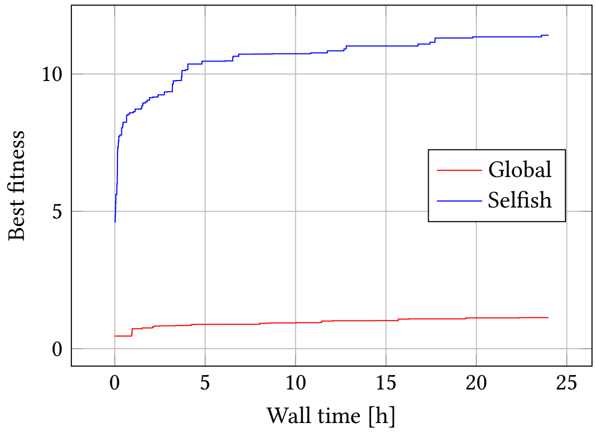

Figure 3: Best fitness (f_{coll} or f_{self}) during the evolution. Values are averaged across the 10 runs.

4.1.2 Homogeneous validation. Figure 4 shows the three indexes (efficiency E, safety S, and average speed V) vs. the number n_{car} of cars obtained in the homogeneous validation. It can be seen that there are slight differences in the average index values between the two variants. In particular, the selfish variant scores a greater efficiency, whereas the global variants scores a greater safety.

As expected, the absolute values of E and S depend on the injected traffic (n_{car}): the heavier the traffic, the lower both indexes. Interestingly, however, the decrease of both efficiency and safety looks more evident for controllers evolved with the selfish variant. This finding suggests that those controllers are less robust to heavy traffic or, from another point of view, more able to exploit the lack of traffic for obtaining better efficiency and safety. The plot in Figure 4 concerning the average speed V appears to corroborate this claim: selfish controllers are always faster (on average) than global controllers, the difference being greater with light injected traffic.

Table 2 shows the median Q_2 and the standard deviation σ, computed across the 100 (10 simulations for each of the 10 controllers of the two variants) values obtained in the homogeneous validation, for the three indexes and for two values of n_{car} corresponding to light (15) and heavy (40) traffic. The table also shows the statistical significance of the results: in particular, for each index and traffic condition, Table 2 reports the p-value obtained with the Mann-Whitney U-test with the null hypothesis that the samples have the same median value. The figures in the table basically confirm what already suggested by Figure 4: however, they also show that the differences among the two methods are not statistically significant. This suggests that, despite the fact that f_{self} does not match to the actual goal which is pursued, driving the evolution with that fitness does not significantly hamper the possibility of achieving that goal. The same data summarized by Table 2 is shown, in the form of box-plots, in Figure 5.

4.1.3 Heterogeneous validation. In the heterogeneous validation, half of the cars were controlled by unskilled controllers, hence providing a test-bed for the controllers being assessed for their robustness w.r.t. the presence of others controllers. Figures 6 and 7

Table 2: Homogeneous validation efficiency E ($\times 10^{-3}$), safety S, and average speed V with $n_{\text{car}} = 15$ (light traffic, top) and with $n_{\text{car}} = 40$ (heavy traffic, bottom).

n_{car}	Index	Collective		Selfish		
		Q_2	σ	Q_2	σ	p-value
15	Efficiency E	3.82	2.73	5.80	3.37	0.256
	Safety S	−4.53	0.52	−4.51	0.50	0.705
	Avg. speed V	6.04	1.38	6.62	0.81	0.130
40	Efficiency E	2.16	1.18	2.94	1.39	0.597
	Safety S	6.57	1.23	7.22	7.06	0.256
	Avg. speed V	4.90	8.83	4.96	0.70	0.650

Table 3: Heterogeneous validation efficiency E ($\times 10^{-3}$), safety S, and average speed V with $n_{\text{car}} = 40$ (heavy traffic).

n_{car}	Index	Global		selfish		
		Q_2	σ	Q_2	σ	p-value
15	Efficiency E	1.86	0.85	2.62	1.82	0.450
	Safety S	−2.08	0.36	−2.08	0.23	0.940
	Avg. speed V	2.81	0.57	2.94	0.33	0.151
40	Efficiency E	1.19	0.74	1.19	1.09	0.880
	Safety S	−3.23	0.49	−3.22	0.26	0.820
	Avg. speed V	2.39	0.36	2.52	0.25	0.130

and Table 3 present the results of the heterogeneous validation as the corresponding figures and table in Section 4.1.2.

In order to provide a more fine-grained view of the results, Figure 6 shows, for each index and each variants, three curves: one with the values computed considering all the cars (solid), one for only the cars controlled by the best controller being assessed (dashed), and one for the remaining cars (dotted).

Overall, it can be seen by looking at the absolute values that in the heterogeneous validation the traffic is in general less efficient and safer, with each car being, on average, slower. From another point of view, the presence of unskilled drivers makes the traffic less fluent.

As in the homogeneous validation, the differences between the two variants are not statistically significant. However, two observations may be made. First, for all the levels of injected traffic, the differences in safety appears negligible: we recall that in the homogeneous validation the global variant looked to result in safer controllers than the selfish variant. Second, with medium traffic the difference in efficiency is clearer in favor of the selfish variant. Despite no sharp conclusions can be drawn, these findings seem to suggest that the controllers evolved with the selfish variant are more robust to the presence of unskilled drivers.

5 CONCLUDING REMARKS

We investigated the feasibility of driving the evolutionary search toward optimizing global, system-wide properties, by promoting selfish behavior, by optimization of local. We compared the two

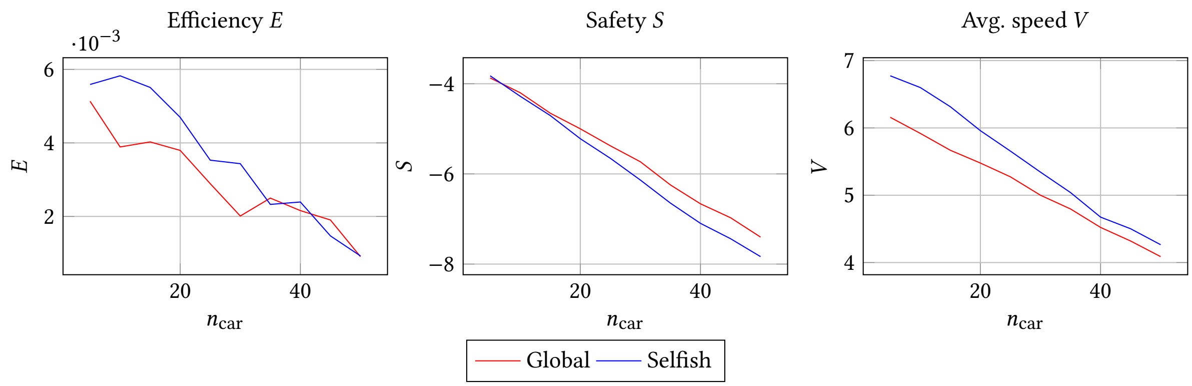

Figure 4: Homogeneous validation efficiency E, safety S, and average speed V, averaged across the 10 simulations, vs. the number n_{car} of cars in the track.

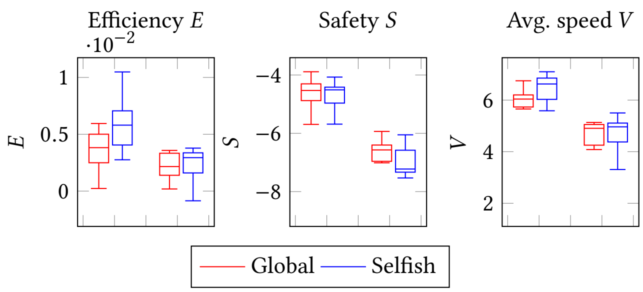

Figure 5: Box-plots of homogeneous validation efficiency E, safety S, and average speed V with $n_{\text{car}} = 15$ (light traffic, left) and with $n_{\text{car}} = 40$ (heavy traffic, right).

approaches by simulation in the context of evolutionary synthesis of a neural-network based car controller for a road traffic scenario. Experimental results showed that the selfish approach is as valid as the global one and suggest that the former may lead to more robust agents than the latter.

REFERENCES

[1] Eric Bonabeau. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences* 99, suppl 3 (2002), 7280–7287.

[2] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. 2013. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence* 7, 1 (2013), 1–41.

[3] Miguel C Figueiredo, Rosaldo JF Rossetti, Rodrigo AM Braga, and Luis Paulo Reis. 2009. An approach to simulate autonomous vehicles in urban traffic scenarios. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*. IEEE, 1–6.

[4] Robert L Goldstone and Marco A Janssen. 2005. Computational models of collective behavior. *Trends in cognitive sciences* 9, 9 (2005), 424–430.

[5] Tobias Holstein, Gordana Dodig-Crnkovic, and Patrizio Pelliccione. 2018. Ethical and Social Aspects of Self-Driving Cars. *arXiv preprint arXiv:1802.04103* (2018).

[6] Ryan Lukeman, Yue-Xian Li, and Leah Edelstein-Keshet. 2010. Inferring individual rules from collective behavior. *Proceedings of the National Academy of Sciences* 107, 28 (2010), 12576–12580.

[7] Eric Medvet, Alberto Bartoli, and Jacopo Talamini. 2017. Road Traffic Rules Synthesis Using Grammatical Evolution. In *European Conference on the Applications of Evolutionary Computation*. Springer, 173–188.

[8] Geoff S Nitschke, Martijn C Schut, and AE Eiben. 2010. Collective neuro-evolution for evolving specialized sensor resolutions in a multi-rover task. *Evolutionary Intelligence* 3, 1 (2010), 13–29.

[9] Stefano Nolfi, Josh C Bongard, Phil Husbands, and Dario Floreano. 2016. Evolutionary Robotics.

[10] Liviu Panait and Sean Luke. 2005. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems* 11, 3 (2005), 387–434.

[11] Julia K Parrish and Leah Edelstein-Keshet. 1999. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science* 284, 5411 (1999), 99–101.

[12] Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation* 10, 2 (2002), 99–127.

[13] Jekanthan Thangavelautham and Gabriele MT D'eleuterio. 2004. A neuroevolutionary approach to emergent task decomposition. In *International Conference on Parallel Problem Solving from Nature*. Springer, 991–1000.

[14] DWF van Krevelen and Geoff S Nitschke. 2008. Neuro-evolution for a gathering and collective construction task. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 225–232.

[15] Chern Han Yong and Risto Miikkulainen. 2001. Cooperative coevolution of multi-agent systems. *University of Texas at Austin, Austin, TX* (2001).

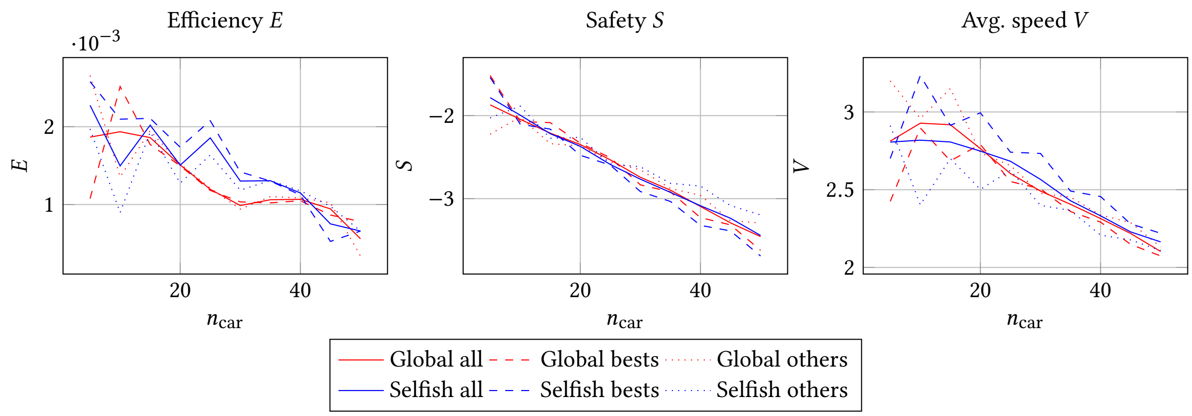

Figure 6: Heterogeneous validation efficiency E, safety S, and average speed V, averaged across the 10 simulations, vs. the number n_{car} of cars in the track.

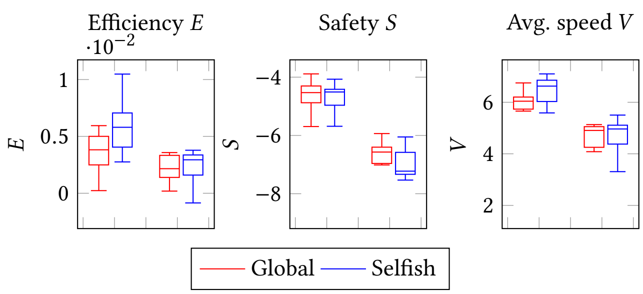

Figure 7: Boxplots of validation efficiency E, safety S, and average speed V with $n_{\text{car}} = 15$ (light traffic).