Multiobjective discovery of driving strategies using the SCANeR Studio

Erik Dovgan
Artificial Intelligence Laboratory
Faculty of Computer and Information Science
University of Ljubljana
Večna pot 113, SI-1000 Ljubljana, Slovenia
erik.dovgan@fri.uni-lj.si

ABSTRACT
This paper presents the SCANeR Studio simulation environment, which is used to design future vehicles by several worldwide companies. In addition, this environment can be used to evaluate autonomous vehicle driving algorithms, although it has several limitations. This paper describes the main environment limitations that are relevant for the evaluation of the Multiobjective optimization algorithm for discovering driving strategies (MODS). Finally, a set of actions for mitigating the SCANeR shortages is given.

Categories and Subject Descriptors
J.7 [Computer Applications]: Computers in other systems

Keywords
driving strategies, multiobjective optimization, SCANeR Studio

1. INTRODUCTION
Autonomous vehicle driving is recently being investigated by many automotive and other companies, e.g., Ford [6], Mercedes-Benz [7], Toyota [12], BMW [5], Audi [8], and Google [13]. Google’s project of autonomous vehicles is led by Sebastian Thrun, former director of the Stanford Artificial Intelligence Laboratory and co-inventor of Google Street View, who won the DARPA autonomous car competition. Several driver assistance systems are already installed in modern vehicles, such as lane assist (see, e.g., Volkswagen [15], Audi [1], and Toyota [14]). In addition, fully autonomous vehicles start driving in urban environments. For example, 100 self-driving Volvo cars will drive on public roads around the city of Gothenburg by 2017 [16]. Till June 2015, the Google autonomous vehicles have driven over 1 million miles encountering 200,000 stop signs, 600,000 traffic lights, and 180 million other vehicles.

The development of an autonomous driving solution usually focuses on the determination of the vehicle surroundings, e.g., other vehicles, obstacles, and pedestrians, in order to increase safety and avoid collisions. However, such a strategy may miss to achieve other objectives that are also important. An objective that has to be taken into account is the reduction of the environmental pollution. This objective is especially important since the awareness of the need to protect our environment is increasing. When driving a vehicle, the pollution reduction is proportional to the reduction of the fuel consumption. An interesting side effect of this is a reduction in traveling cost, although this is countered by an increase in traveling time. It is clear that heavily reducing fuel consumption is not optimal because it leads to unacceptably long traveling time. Therefore, several objectives need to be taken into account simultaneously when constructing a driving strategy.

To obtain autonomous vehicle driving, real-time driving strategies have to be applied by the autonomous vehicle driving algorithms. These strategies have to select the best control action at each step with respect to the current vehicle and route state, e.g., position on the route, position of neighbor vehicles, velocity of neighbor vehicles etc. The best control action can be selected by implementing an algorithm that optimizes various objectives, such as the traveling time, the fuel consumption, the distance to other vehicles for collision avoidance etc. Since several objectives have to be simultaneously taken into account, it is preferable to use the multiobjective approach when developing the optimization algorithm, since it is in general the case that multiobjective algorithms exhibit advantages over single-objective algorithms (since they enable, e.g., better exploration of the multiobjective search space).

To use the autonomous vehicle driving algorithms in practice, the proper evaluation of these algorithms is of key importance. Since untested algorithms cannot be simply deployed in autonomous vehicles and tested in real environment, e.g., city roads, the evaluation has to be performed in the near-real-life driving simulation environment. Typically, the best existing driving simulation environments are used by the worldwide automotive companies. This paper examines the SCANeR Studio simulation environment [9], which is used by several worldwide companies such as Renault, PSA Peugeot Citroën and VOLVO Trucks [10].

The goal of the ongoing research is to enhance the existing Multiobjective optimization algorithm for discovering driving strategies (MODS) [2, 3, 4] to take into account real-life driving situations and to evaluate it with the SCANeR Studio. To this end, MODS will have to be significantly adapted due to limitations of the SCANeR Studio.

The paper is further organized as follows. Section 2 describes the existing Multiobjective optimization algorithm for dis-
covering driving strategies. The SCANeR Studio and its limitations are presented in Section 3. Section 4 presents the ongoing and future enhancements of the MODS algorithm. Finally, Section 5 concludes the paper with the summary of work.

2. MULTIOBJECTIVE OPTIMIZATION ALGORITHM FOR DISCOVERING DRIVING STRATEGIES

Multiobjective optimization algorithm for discovering driving strategies (MODS) [2, 3, 4] is a two-level multiobjective optimization algorithm that finds a set of nondominated driving strategies with respect to two conflicting objectives: traveling time and fuel consumption. The lower-level algorithm is based on a deterministic breadth-first search, driving prediction and nondominated sorting, and searches for nondominated driving strategies. The search is performed by simulating vehicle driving by route steps. It starts with a single driving strategy with no control actions. If for a route step the driving strategy does not define the control action, the driving strategy is cloned for each possible control action and the obtained driving strategies are stored in the set of driving strategies. When the control action is defined, it is used to predict the vehicle driving for a predefined number of prediction steps. Due to the cloning, the number of driving strategies grows exponentially. To maintain a constant number of driving strategies at each route step, fast nondominated sorting is used to select the most promising driving strategies.

The upper-level algorithm is an evolutionary algorithm that optimizes the input parameters for the lower-level algorithm. This algorithm applies evolutionary mechanisms, i.e., selection, crossover and mutation, to the sets of input-parameter values through generations and maximizes the hypervolume covered by the driving strategies found by the lower-level algorithm.

MODS was implemented in two variants. The original MODS algorithm minimizes the traveling time and the fuel consumption [2, 3], while the enhanced version, called MOCDS, optimizes also the driving comfort [4].

Currently, the MODS algorithm is integrated and was evaluated in a vehicle driving simulator that includes data from real-world routes and a black-box vehicle simulator. The results show that MODS finds better driving strategies than existing algorithms for discovering driving strategies. However, MODS has some limitations. For example, it does not take into account neighbor/other vehicles on the route, it does not produce human-like driving strategies, and it does not find (good) driving strategies in real time. To overcome these shortages, the ongoing research consists of design and implementation of the enhanced MODS algorithm, and its deployment in the near-real-life environment, i.e., the SCANeR Studio simulation environment.

3. SCANeR STUDIO SIMULATION ENVIRONMENT

The SCANeR Studio [9] is a modular driving simulation environment, developed by the OKTAL company [11]. It simulates the vehicle driving behavior and the behavior of various entities in the environment. To this end, it consists of several modules such as:

- Complex dynamic vehicle models simulating the behavior of every component of a real vehicle
- Simple models of autonomous traffic vehicles simulating the behavior of other vehicles on the road
- Models for simulating the behavior of pedestrians

The behavior of these models can be predefined, controlled online with the scripting language or managed through APIs using custom algorithms. In addition, complex vehicle models can be controlled by the driver through the acquisition modules that can be connected to keyboards, gaming steering wheels etc. Vehicle driving is presented to the driver using visual and sound modules, and the module that controls dynamic platforms (if present). Moreover, the models involved in the simulation can be controlled/executed with either SCANeR user interface or programs that control the simulation, i.e., the supervisors.

The SCANeR Studio enables to create a complete driving simulation environment through the usage of five dedicated modes of the graphic interface:

- Terrain mode: Road network creator allowing the rapid creation of realistic road networks that are usable directly in the simulation
- Vehicle mode: Tool for fine-tuning and study of dynamic vehicle models
- Scenario mode: Driving simulator scenario editing tool
- Simulation mode: Simulation supervision tool
- Analysis mode: Detailed graphical analysis tool

Although The SCANeR Studio is a powerful tool for evaluating autonomous vehicle driving algorithms, it introduces some limitations for the development of these algorithms. When taking into account the MODS algorithm, the SCANeR Studio limitations presented Section 3.1 are the most critical ones.

3.1 Limitations of the SCANeR Studio

Limited implementation of complex dynamic vehicle models

The SCANeR Studio contains several types of complex dynamic vehicle models such as cars, trucks and buses. The instances of these models can be controlled only with throttle, braking and clutch pedals, gearbox, steering wheel etc. On the other hand, if an autonomous vehicle driving algorithm controls the vehicle by setting vehicle’s velocity, such algorithm cannot be evaluated with complex vehicle models.

The SCANeR Studio contains also a large set of various simple vehicle models. These models define some vehicle parameters such as length, weight etc., but do not simulate engine behavior in details. For example, they do not
simulate fuel consumption or engine limitations. Since no engine limitation is simulated, these simple models enable to instantaneously change the vehicle velocity without limitations, which results in unrealistic driving behavior. The advantage of these algorithms is that they can be controlled by autonomous vehicle driving algorithms that set vehicle's velocity.

**Limited offline simulation**

The SCANeR Studio enables to execute the simulation in an offline mode that schedules the execution of the SCANeR and other modules, involved in the simulation, with respect to their running frequency, and then executes them with respect to the schedule without any sleep time. This enables to execute the simulation as fast as possible, which is specially suitable when the autonomous vehicle driving algorithms control the vehicle behavior and no user interaction is required.

The offline simulation can, however, be used only when user executes such a simulation thought the SCANeR user interface. On the other hand, when the simulation has to be executed by a supervisor program, the offline simulation cannot be used. This represents a significant limitation for optimization programs that need to test various parameter settings of an autonomous vehicle driving algorithm, where each setting has to be evaluated with the execution of the same simulation scenario.

**Driving prediction is inapplicable**

Several autonomous vehicle driving algorithms use the driving prediction technique to construct the driving strategy and select the best control action at each step. The usage of driving prediction requires to periodically (ideally at each step) stop the driving simulation, start the driving prediction from the current vehicle and route state possibly multiple times if various control actions have to be tested, and then apply the obtained information during prediction in order to continue with the driving simulation. To execute such a procedure, the main simulator has to simulate the driving, while additional simulators have to be periodically executed to predict the driving. Such a configuration is not supported by the SCANeR Studio, since SCANeR requires to execute only one instance of SCANeR modules simultaneously. Otherwise, conflicts can appear in the communication between modules if several instances of the same module are active simultaneously.

**Synchronization issues**

The modules of the SCANeR Studio are implemented and executed as independent processes that communicate between them through network. Since they are executed as independent processes, their actual execution depends on the hardware and OS specifications, e.g., the number of processor cores that can simultaneously execute various processes. Consequently, the sequence of messages sent between modules is not deterministic, which is also due to the network latency. For example, messages from vehicle model containing the current vehicle velocity may sometimes be delayed with respect to messages for autonomous vehicle driving algorithm containing control actions, which may result in delayed reaction of the autonomous vehicle driving algorithm.

The main issue of such behavior is the fact that the simulation results are not deterministic, i.e., the same scenario with the same SCANeR modules does not always produce the same driving behavior and consequently the same traveling time, fuel consumption etc. Therefore, an instance of the autonomous vehicle driving algorithm has to be evaluated several times in order to statistically evaluate its quality.

The synchronization issues are, however, present only in the online simulation mode. On the other hand, the offline simulation mode does not have these issues, since it schedules the execution of the SCANeR modules in advance and then executes them with respect to the predefined, deterministic schedule. Consequently, the sequence of messages sent between modules is also deterministic.

### 3.2 Adaptation of MODS to the SCANeR Studio

The limitations of the SCANeR Studio significantly influence the enhancement of the MODS algorithm. Since the MODS algorithm controls the vehicle by setting vehicle's velocity, it requires to use SCANeR's simple vehicle models. Due to the fact that these models do not simulate engine behavior in details, such engine behavior will have to be additionally implemented. This will require the implementation of, for example, wheel friction, aerodynamic, slope friction, inertial, maximum torque, specific fuel consumption, and maximum engine moving functions, in order to limit the vehicle inertial force and calculate the fuel consumption.

The MODS algorithm will require the usage of offline simulation in order to execute the multiobjective optimization algorithm. Ideally, the multiobjective optimization algorithm should be implemented as a supervisor program. Due to the fact that a supervisor algorithm cannot control the offline simulation, the simulation will have to be controlled with HTTP POST calls. Although controlling SCANeR with POST calls is not optimal, this is the only mode that can be currently applied for the offline simulation.

Driving prediction is one of the key mechanisms of the MODS algorithm. Since this prediction is inapplicable in the SCANeR Studio, the MODS algorithm will have to be significantly redesigned. To compensate the missing driving prediction, a simple prediction approach will have to be added to the MODS algorithm, e.g., a set of rules that predict the future collision time.

MODS also requires exact evaluation of the driving strategies. Since this can be achieved only by the offline simulation mode, only this mode will be used. Consequently, the driving strategies obtained with the online simulation mode will have to be re-evaluated with the offline simulation mode before the comparison with the MODS driving strategies.

### 4. ENHANCEMENTS OF THE MODS ALGORITHM

The MODS algorithm is going to be significantly enhanced in order to take into account real-life driving situations and to be evaluated with the SCANeR Studio. Currently, MODS is being adapted in order to control the vehicle's velocity. In addition, the enhanced algorithm uses only the current
and the next velocity limits as hypercube dimensions of the driving strategies [3]. The other attributes, such as the route inclination, are added to the formulas that change the target velocity. In the future, the hypercube dimensions will be redefined by taking into account the results of the analysis of the human driving strategies.

To adapt to the SCANeR Studio limitations, the prediction of the lower-level MODS is not used anymore. Nevertheless, a simple prediction will be added to the MODS algorithm if needed. The lower-level MODS will be also enhanced to simulate the car following, lane changing, and vehicle overtaking behavior.

The upper-level MODS is going to be adapted to take into account the enhanced definition of the hypercubes. The new parameters of the driving strategies, which will be stored in hypercubes, will include the vehicle velocity, acceleration, deceleration, duration of acceleration, duration of deceleration, etc. In addition, the upper-level MODS will also be enhanced to search for parameter values for the car following, lane changing, and vehicle overtaking behavior.

5. CONCLUSIONS

This paper presented the SCANeR Studio simulation environment and its limitations that are relevant for the enhancement of the Multiobjective optimization algorithm for discovering driving strategies (MODS) and its deployment in this Studio. Four main limitations were described: limited implementation of complex dynamic vehicle models, limited offline simulation, inapplicability of driving prediction, and synchronization issues. The analysis of MODS and SCANeR Studio shows that MODS will need several adaptations before the evaluation with the SCANeR Studio, e.g., the implementation of engine behavior and the algorithm redesign to remove the driving prediction from the algorithm and adapt it accordingly.

6. ACKNOWLEDGMENTS

The work presented in this paper was in part funded by the NERVtech, raziskave in razvoj, d.o.o., which also provided the SCANeR Studio simulation environment. In addition, this work was also in part funded by the Slovenian Research Agency under research project Z2-7581.

7. REFERENCES


