A Multi-Objective Evolutionary Algorithm for the Tuning of Fuzzy Rule Bases for Uncoordinated Intersections in Autonomous Driving

E. Onieva, U. Hernandez-Jayo, E. Osaba, A. Perallos, X. Zhang

Deusto Institute of Technology (DeustoTech), University of Deusto, Bilbao 48007, Spain.
email: enrique.onieva@deusto.es, unai.hernandez@deusto.es, e.osaba@deusto.es, perallos@deusto.es, xiao.zhang@deusto.es

Abstract

This paper focuses on the application of Multi-Objective Evolutionary Algorithms (MOEA) to develop Fuzzy Rule-Based Systems (FRBS) dedicated to manage the speed of an autonomous vehicle in an intersection scenario.

The main particularity with other intersection scenarios is that the autonomous vehicle is approaching an intersection being crossed by a row of manual vehicles, which do not pay attention to the presence of the former, thus making coordination impossible. In this case, the autonomous vehicle is the only responsible of adapting its speed to the state of the rest of vehicles, with the aim of finalizing the maneuver both in a safe and efficient way.

The problem presents some specific restrictions that make it very particular and complex, because of the large time requirements needed to consider multiple criteria (which enlarge the solution search space) and the long computation time required in each evaluation. In addition, due to the high number of variables involved, the complexity of the scenario is considerable.

In this work, a MOEA is proposed to obtain more compact and efficient FRBS. The proposal is based on the well-known Strength Pareto Evolutionary Algorithm 2 (SPEA2) technique, but uses different mechanisms for guiding the search towards the desired Pareto zone. The MOEA uses specific operators to deal with the problem, as a method to *inherit* fitness values among generations, obtaining so, that is necessary only to execute the individuals in only one scenario per generation to obtain FRBSs that work fine in many situations. In addition, the most important rules are identified in each FRBS, with the aim of realizing balanced crossovers.

Keywords: Intelligent transportation systems, Autonomous driving, Fuzzy logic control, Multi-objective evolutionary algorithms, Fuzzy rule-based systems, Multi-objective evolutionary fuzzy systems

1. Introduction

A substantial fraction of vehicle collisions occur at intersections. In particular, vehicle collisions at intersections are between 25% and 45% of all of them [37]. Since intersections represent a very small portion of the roadway, this is considered a disproportionate amount. Due to that, intersection safety remains a challenge both for Advanced Driver Assistance Systems (ADAS) and autonomous driving [36].

Statistical studies of the causes of accidents at intersections have shown that 90% of them are due to driver error [23]. The most common errors are perception failures (e.g. inattention), misunderstanding (e.g. misjudging the intentions of another driver), and wrong decision (e.g. incorrect maneuver).

Intelligent Transportation Systems (ITS) provide information and robotic techniques to achieve safe and efficient driving. In the automotive industry, sensors are mainly used to give information to the driver, or to advise him about the presence of a dangerous situation [44]. In some cases, they are connected to a computer that performs some guiding actions, attempting to minimize injuries and prevent collisions [4].

ITS can play a key role in order to avoid these hazardous situations, using two main technical structures: first, a communication or sensorial platform is required in order to allow the vehicles to interchange (or know) information about their position and driving parameters (speed, acceleration, direction). Secondly, a set of intelligent management algorithms, defined to minimize accidents advising the driver about the presence of a dangerous area, or acting over the vehicle actuators [32].

Intersection collision warning is a cooperative awareness application for improving active road safety, which can be deployed in both controlled and uncontrolled intersections [33, 43].

In this topic many research works have been carried out. Most of them assume a full control of the intersection, usually in two ways: (i) by controlling speeds of all vehicles involved (or sending speed references to all of them) [25, 28, 17]; (ii) by controlling intersection's signalization [40, 22]. In particular, [25] uses a dynamic signal controller, [28] presents a fuzzy logic based system that coordinates a group of autonomous vehicles approaching to the same intersection and [17] develops a formal methodology to control two vehicles approaching to an intersection. [40] controls the traffic lights by fuzzy logic methods, while in [22], the elements deployed in the road are used as sensors to gather information about the traffic flow, and feed a fuzzy system that defines the optimal signal status. In terms of vehicle control, intersection manouvres assume cooperation between approaching vehicles, in order to control both speeds [24, 12], or, when no cooperation exists, the autonomous vehicle takes the role of give way to the approaching vehicle [46, 27] or, in more advanced systems it calculates its own precedence with respect with the approaching vehicle [42, 3].

Fuzzy logic [47] is usually used in the development of systems designed to deal with the complexity derived from traffic situations [19]. It allows the actions and decisions involved to be described in terms of simple rules, as well as driving related maneuvers, can be described easily with rules. For example: *if a vehicle is stopped in front of me and I am driving very fast then there exists a collision risk*. Fuzzy logic has proven to be a technique well suited to the treatment of all kinds of transportation problems [29, 31].

Genetic Algorithms (GAs) are stochastic search techniques inspired by the principles of natural selection and evolution of species [18, 16]. GAs are popular research subjects, since they can deal with complex engineering problems which are difficult to solve by classical methods [21]. GAs have been also widely used in literature for Fuzzy Rule-Based Systems (FRBS) tuning [2, 30].

Multi-Objective Evolutionary Algorithms (MOEAs) are one of the most active research areas in the field of evolutionary computation, because they are population-based algorithms being capable of capturing a set of non-dominated solutions in a single run. A large number of this kind of algorithms have been proposed in the literature [8]. Among them, the Non-Dominated Sorting Genetic Algorithm (NSGA) [11] and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [48] are well known and frequently used MOEAs. Finally, with respect to MOEAs applied to FRBSs, the interpretability-precision trade-off approach [7] is one of the most attractive field, since it allows to obtain a balance between the precision and complexity of the obtained FRBSs [6].

In this contribution, we propose an effective and efficient SPEA2 based strategy that incorporates specific mechanisms, in order to better optimize a FRBS capable of generating autonomously speed references to an autonomous vehicle that approaches to an intersection situation where a group of vehicles is crossing. Since cooperation from the group of vehicles can not be assured, it is the responsibility of the autonomous system to finalize the maneuver without risk. Strategy is implemented with the aim of satisfying the following restrictions: (i) to guide the vehicle to cross the intersection without any collision with other vehicle; (ii) to do it in the less possible time; (iii) to be able to deal with as many situations as possible, defined in terms of speed of vehicles, gaps between them, etc.

To deal with the problem, SPEA2 is provided with specific and original operators and mechanisms:

- Execution of the FRBSs in a different scenario in each iteration of the process, in order to test them under a wide set of conditions.
- A mechanism to infer the generality of the FRBS, based on its behavior over the actual and previous scenarios.
- A crossover operator that uses information coming from the simulation, in order to identify the most important rules in each FRBS and to combine them in a balanced way.
- An initialization operator to generate an initial population with general individuals which cover a high number of situations.
- Membership function codification variable over the time allows to interpret several similar situations as only one, reducing so, the number of input variables needed.

The paper is structured as follows: Section 2 describes some preliminary concepts, Section 3 details the problem to be solved, Section 4 explains the multi-objective evolutionary approach implemented, Section 5 shows the experimentation carried out and the analysis of results. Finally, Section 6 outlines the conclusions and further research.

2. Preliminaries

Since the proposed method is based on the well known SPEA2 [48], it is introduced in the Section 2.1. In addition, in Section 2.2, some of the basis about the fuzzy logic and control are presented.

2.1. SPEA2

SPEA2, suggested by Zitzler in [48], is one of the most popular techniques for solving multi-objective problems.

The algorithm was designed to polish the deficiencies of its predecessor, the Strength Pareto Evolutionary Algorithm (SPEA) [49]. SPEA uses a population and an archive, in which they are kept the non-dominated individuals. If the size of the archive exceeds a predefined limit, further archive members are deleted by a clustering technique. Afterward, fitness values are calculated by summing the strengths of non-dominated individuals that dominate a certain individual.

In contrast with the first version, SPEA2 introduced the following innovations, in order to address some inefficiencies of the initial approach:

1. The fitness is assigned taking into account a raw fitness and a density value, instead of uniquely the rank.
2. To guide the search more efficiently, SPEA2 uses the nearest neighbor density estimation technique.
3. To support the diversification in the solution space, SPEA2 has an enhanced archive truncation method that removes solutions in overcrowded regions.

SPEA2 also uses a population and an archive, henceforth P_t and $\overline{P_t}$, with constant sizes N and $\overline{N}$. The fitness assignment strategy takes into account both dominating and dominated solutions for each individual. In detail, each individual i in $P_t \cup \overline{P_t}$ is assigned a strength value $S(i)$, which is calculated as:

$$S(i) = \left\| \{ j | j \in P_t \cup \overline{P_t} \;\&\; i \succ j \} \right\| \tag{1}$$

where $\|\cdot\|$ represents cardinality, and $\succ$ corresponds to the dominance relation. Based on $S(i)$, a raw fitness $R(i)$ is calculated as the sum of the $S(j)$ of all individuals it dominates. Stated in a formal way:

$$R(i) = \sum_{j \in P_t \cup \overline{P_t}, j \succ i} S(j) \tag{2}$$

The final fitness value is assigned by adding the density value $D(i)$, which is calculated as:

$$D(i) = \frac{1}{\delta_i^k + 2} \tag{3}$$

, where δ_i^k denotes the $k-th$ nearest distance for the $i-th$ individual in $P_t \cup \overline{P_t}$. k is usually set to $\left\lfloor \sqrt{N + \overline{N}} \right\rfloor$. In Figure 1 an example of $R(i)$ and $D(i)$ calculation is presented. The final fitness value is assigned adding both values:

$$F(i) = R(i) + D(i) \tag{4}$$

The archive is constructed and updated by copying all non-dominated individuals in $P_t \cup \overline{P_t}$ into a temporary archive $\overline{P_{t+1}}$. If the size of this temporary archive differs from $\overline{N}$, individuals are added or removed as follows:

- If the archive is too small ($|\overline{P_{t+1}} < \overline{N}|$), additions are made selecting the individuals in $P_t \cup \overline{P_t}$ with lower F (Equation 4).
- In case the archive is too large ($|\overline{P_{t+1}} > \overline{N}|$), individuals with higher D (Equation 3) are iteratively removed.

This procedure is done to ensure a wide diversity of the objective space in the archive. Finally, Algorithm 1 presents the main steps to implement SPEA2.

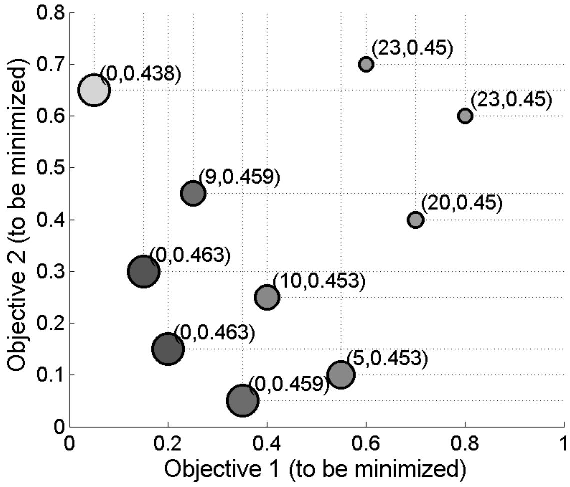

Figure 1: The SPEA2 raw fitness and density calculation. Labels indicate $(Raw\ Fitness, Density)$; Markers are presented in different sizes (the biggest, the less raw fitness) and colors (the lightest, the less density) for a better understanding.

2.2. *Fuzzy Control*

Fuzzy logic, suggested by Zadeh in [47]. It is a powerful and widely used technique to include reasoning to decision-making problems. It is a rigorous mathematical field which is based on uncertainty and vagueness [45], being able to model these features analytically.

One important component of fuzzy logic is the setting of expressions in a similar way to *the speed is high*. These rules, which are used in the natural language, are transformed into analytical expressions through the *fuzzyfication* process. A fuzzy set is characterized by its Membership Function (MF):

$$\mu_{MF}(x) : X \rightarrow [0, 1] \tag{5}$$

The fuzzy set defined by MF can be modeled as a set of ordered pairs as:

$$MF = \{(x, \mu_{MF}(x))|x \in X\} \tag{6}$$

For practical purposes, MFs are usually formulated with idealized shapes, being triangular, trapezoidal, or Gaussian the most common ones. In our case, we use triangular MFs to codify input variables, which can be coded by three real values (a, b, c); the calculation of membership degree is done as:

$$\mu(x, \{a, b, c\}) = \begin{cases} \frac{x-a}{b-a}, & if\ (x \in [a, b]). \\ \frac{b-x}{c-b}, & if\ (x \in [b, c]). \\ 0, & otherwise. \end{cases} \tag{7}$$

Figure 2 depicts an example of fuzzy sets for measuring the speed of a vehicle. In the Figure, the four triangular MFs are associated to the terms $\{stop, slow, moderate, fast\}$.

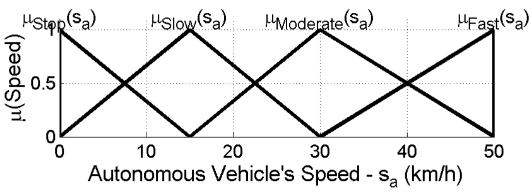

Figure 2: Example of fuzzy sets for measuring the speed of a vehicle.

Data: N (population size)
$\overline{N}$ (archive size)
T (maximum number of generations)
Result: $\overline{P}$ (non-dominated set)
$t \leftarrow 0$
$P_0 \leftarrow$ Initial Population
$\overline{P_0} \leftarrow \phi$
while $t < T$ **do**
  Calculate fitness of individuals in P_t and $\overline{P_t}$
  $\overline{P_{t+1}} \leftarrow$ non-dominated individuals in $P_t \cup \overline{P_t}$
  if $t \geq T$ **then**
    return $\overline{P_{t+1}}$ and stop
  end
  while $|\overline{P_{t+1}}| > \overline{N}$ **do**
    Delete individual with higher D (Eq. 3) from $\overline{P_{t+1}}$
  end
  while $|\overline{P_{t+1}}| < \overline{N}$ **do**
    Insert individual with lower F (Eq. 4) from $P_t \cup \overline{P_t}$
  end
  Apply selection operator over $\overline{P_{t+1}}$ to fill the mating pool
  Apply crossover and mutation over the mating pool and set P_{t+1} to the resulting population
  $t \leftarrow t + 1$
end

Algorithm 1: Pseudocode for SPEA2

In the present work we use a zeroth-order Takagi-Sugeno-Kang (TSK) fuzzy system, where rules can be defined as:

$$R_i : IF\ (x_{(1)}\ is\ MF_{(j_1,1)})\ \{AND|OR\}\ \ldots\ (x_{(N)}\ is\ MF_{(j_N,N)})\ THEN\ y\ =\ Y^i$$

where $x_{(n)}$ is the $n\!-\!th$ input variable used in the rule antecedent. $MF_{(j,n)}$ is the $j\!-\!th$ MF of the $n\!-\!th$ input variable used in the rule, and Y_i is a numerical value representing the location of the singleton that acts as rule consequent.

The t-norm *minimum* and the t-conorm *maximum* are used to implement the *AND* and *OR* operators. Mamdani-type inference [26] is used, and the *defuzzification* operator is the *weighted average*. Therefore, the final output value is calculated as follows:

$$Y = \frac{\sum Y_i \cdot \mu_i}{\sum \mu_i} \tag{8}$$

where μ_i represents the degree of truth of the $i - th$ rule, and Y_i is the value of the singleton inferred by the $i - th$ rule.

Sugeno proved in [38] that a fuzzy controller modeled with singletons as consequents is a special case of a fuzzy controller modeled with trapezoidal ones, being able to do almost everything the latter can do.

3. Proposed Scenario

In this paper, a scenario consisting of a crossroad without traffic signals is proposed. In it, a group of vehicles is approaching at the same time. One of them implements an autonomous strategy, while the other ones are manually driven, as shown in Figure 3. In the basic case, the autonomous vehicle must follow a simple priority strategy, letting pass to the manual ones, since they are approaching by the right. This is a basic rule of the traffic law in Spain:

Article 57 of chapter 3 of the Spanish Road Circulation Code reads as follows: In the absence of signals that regulate the priority, drivers are obliged to yield the pass to vehicles approaching by their right.

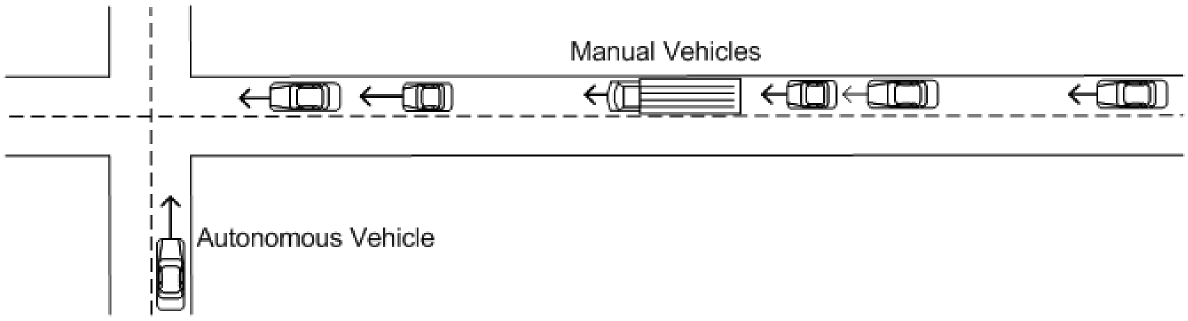

Figure 3: Proposed scenario. An autonomous vehicle approaches an intersection with crossing traffic.

Despite this, manually driven vehicles have no obligation to cooperate with the autonomous one, so, the latter has to independently adapt its speed to the situation as an human driver would do (looking for the first opportunity of crossing with safety and without interrupting to the rest of vehicles). For this reason, the aim of the present work is to optimize the maneuvering of the autonomously driven vehicle. For this purpose, autonomous vehicle is supposed to receive information about the positioning and speed of the manually driven ones. An on-board fuzzy system will be in charge of processing the information to determine the most suitable speed to finish the maneuver without risk. On the other hand, when there are no vehicles approaching the intersection, autonomous vehicles must drive normally.

To make the autonomous vehicle be able to deal with this situation, we define a two-layer control structure. First, a parametrized FRBS calculates the proper speed for maneuvering in a safe and efficient way. Then, the previous module transmits orders to the vehicle's actuators (throttle and brake), in order to adjust the speed.

Section 3.1 will describe in detail the variables used by the FRBS. Section 3.2 presents the longitudinal vehicle model used to simulate the evolution of the autonomous vehicle's speed.

3.1. Variables used by the Fuzzy Rule Based System

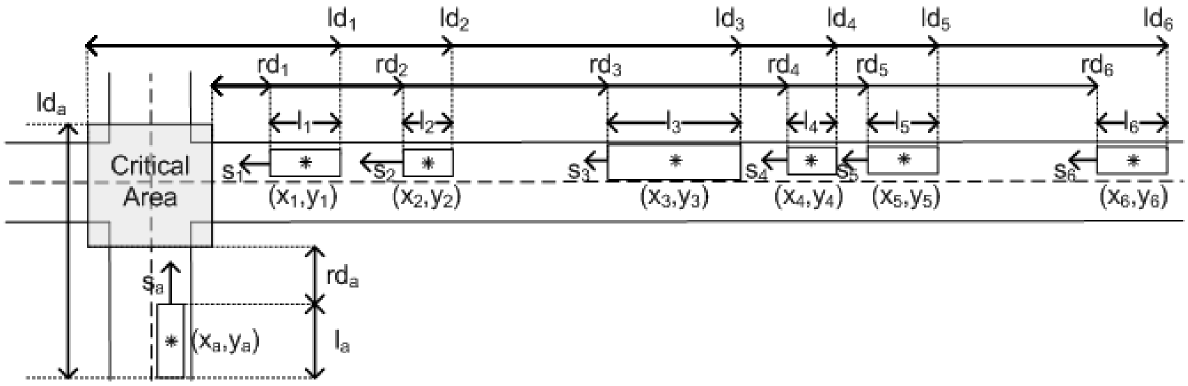

Figure 4: Graphical representation of the scenario's involved variables. Example with 6 manual vehicles.

We take the assumption that the autonomous vehicle is able to receive coordinates and lengths of all the vehicles approaching the intersection, as well as it knows its own ones. Henceforth, (x_a, y_a, l_a) will denote the position and length (in meters) of the autonomous vehicle, while (x_i, y_i, l_i) represents the same for the $i-th$ manually driven one. Note that once the first manual vehicle leaves the intersection, the second one starts to be considered as the first, and so on. In addition, the autonomous vehicle knows the speeds of all the vehicles involved in the scenario, denoted as s_a for the autonomous and s_i for the $i-th$ manual one.

This information can be received by the autonomous vehicle by communications, both vehicle-to-vehicle or vehicle-to-infrastructure [14, 35], or by it own sensing, which can be based in cameras, radar or lidar, among others [20].

Another parameter taken into account is that we call *critical area*. This *critical area* is defined by a square centered in the middle of the crossroad, with 5m on each side of this crossroad. The objective of this area is to prevent the autonomous vehicle to enter in it while there is a manual one inside.

With this information, the autonomous vehicle is able to calculate the distances to reach and leave the *critical area* for each vehicle, hereafter rd_x and ld_x, being $x = a$ for the autonomous driven and $x = i$ for the $i - th$ manual one. In the same way, $rt_x = rd_x/s_x$ and $lt_x = ld_x/s_x$ are defined as the time (in seconds) needed for a vehicle to reach and leave the *critical area*.

A graphical representation of all the variables involved in the scenario is shown in Figure 4, where the *critical area* is defined by the gray square. In figure, we can see that between each two vehicles exists an gap. In this regard, the inter distance between the vehicle i and $i + 1$ will be represented by I_i. This parameter can be easily calculated in different ways from known information:

$$I_i = rd_{i+1} - rd_i - l_i = ld_{i+1} - ld_i - l_{i+1} \tag{9}$$

, for practical purposes, if I_i divided by the speed of the leader vehicle does not excess a certain threshold (T_{inter}), they both will be considered as one single vehicle, considering the speed of the leader as the speed of the group. This means that if the time window available to cross between two manual vehicles is less than T_{inter}, they both are considered as a single vehicle. With this consideration, the scenario shown in Figure 4 can be simplified as presented in Figure 5.

Therefore, from the point of view of the controller, when vehicle i is refereed, it must be understood as the corresponding group of vehicles. On the other hand, the grouping heuristic is done in a continuous way, so groups can be split and merged if vehicles alter their speeds or their I_i.

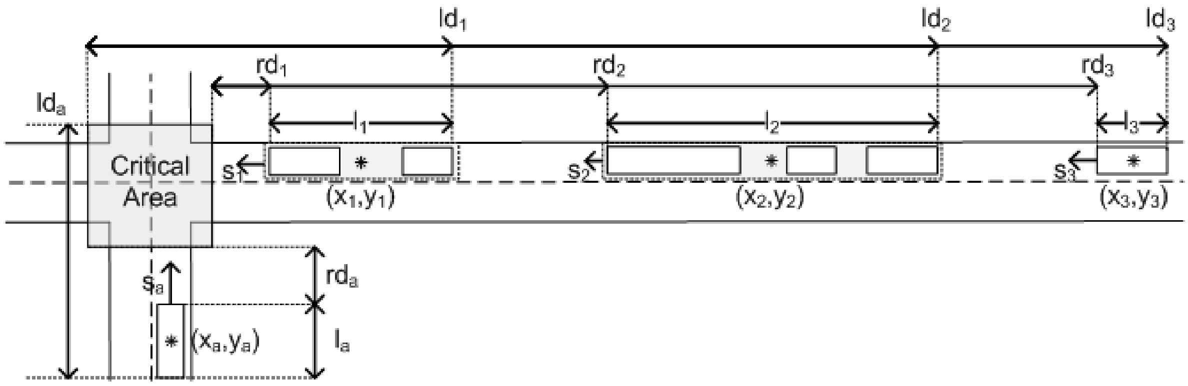

Figure 5: Graphical representation of the scenario's involved variables after grouping vehicles.

It is important to note that in this work, it is used a simplified version of an intersection scenario where a group of manual vehicles are approaching by the right side of the autonomous one without turn intentions. More complex scenarios can be converted to one of the presented type by: (i) calculating the ld_i value along the desired trajectory of the manual vehicles, (ii) not considering vehicles with trajectories that do not interfere with the one of the autonomous, (iii) by considering all the vehicles approaching by the same lane, (iv) by implementing an adaptive cruise control policy with the preceding vehicle (if exists). These modifications are represented in a graphical way in Figure 6. Conversions (i) and (ii) assume the autonomous vehicle knows the turn intentions of the manual ones, which would imply a more complex instrumentation in the vehicles.

3.2. Longitudinal behavior of the autonomous vehicle

The longitudinal behavior of the autonomous vehicle can be approximated –for slowly varying dynamics and on a flat surface– by the second-order transfer function:

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\eta\omega_n + \omega_n^2} \tag{10}$$

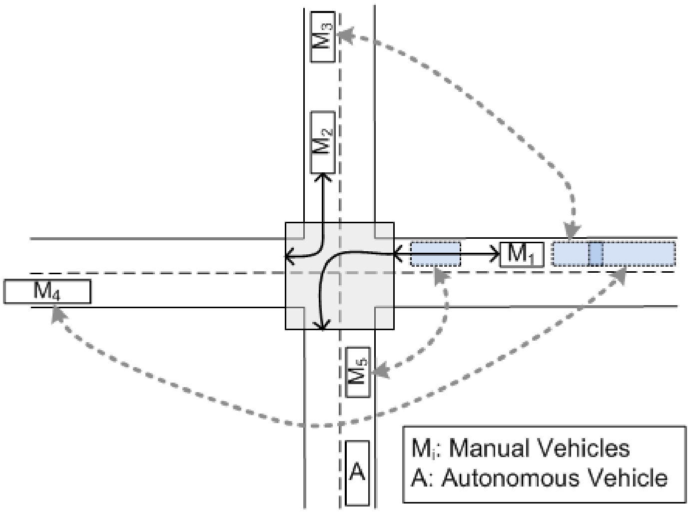

Figure 6: Example of how the proposed scenario can be extended to more complex situations. M_1 is going to turn left, so its leaving distance is calculated along the trajectory; M_2 does not interfere with the autonomous vehicle, so it is not considered. M_3, M_4 and M_5 are considered in the same lane that M_1. In addition, the autonomous vehicle must implement an adaptive cruise control policy with M_5

where $K = 25.14$, $\eta = 160$, and $\omega_n = 55.87$ (see [41] for further details).

A Proportional-Integral (PI) controller was implemented to attain the target speeds provided by the *target speed determiner*. This well-known control technique [1] is not only easy to implement, but also allows to describe its behavior in the Laplace domain as $C(s) = K_P E(s) + K_I \frac{E(s)}{s}$, being $E(s)$ the tracking error, and $K_P = 0.3$, $K_I = 0.1$ the control gains. As a result, the closed-loop system dynamics are described by the continuous transfer function:

$$H(s) = \frac{C(s)G(s)}{1 + C(s)G(s)} = \frac{2.35 \cdot 10^4 s + 2.35 \cdot 10^4}{s^3 + 1.79 \cdot 10^4 s^2 + 2.67 \cdot 10^4 s + 2.35 \cdot 10^4} \tag{11}$$

The system can be discretized using the bilinear transform $s \rightarrow \frac{2}{T}\frac{z-1}{z+1}$, with the resulting discrete approximation of the vehicle's longitudinal dynamics being

$$H(z) = \frac{a_2 z^2 + a_1 z^1 + a_0}{z^3 + b_2 z^2 + b_1^z + b_0} \tag{12}$$

being $a_0 = -5.467 \cdot 10^{-5}$, $a_1 = -0.2041$, $a_2 = 0.2495$, and $b_0 = 0$, $b_1 = 0.7421$, $b_2 = -1.697$.

Finally, the discrete transfer function of Equation (12) can be rewritten as a linear constant coefficient difference equation:

$$s_a(t_k) = a_2 s_{ref}(t_{k-1}) + a_1 s_{ref}(t_{k-2}) + a_0 s_{ref}(t_{k-3}) - b_2 s_a(t_{k-1}) - b_1 s_a(t_{k-2}) - b_0 s_a(t_{k-3}) \tag{13}$$

where $s_{ref}(t_k)$ and $s_a(t_k)$ are the reference and actual velocities at instant t_k.

To illustrate the behavior of the vehicle speed using the implemented model, in Figure 7 some variations in the speed of the autonomous vehicle under different initial/reference speed configurations are shown.

4. Multi-objective Evolutionary Proposal

In this section, we explain the algorithm implemented to deal with the proposed problem. As mentioned in Section 2, the proposed method is based in the SPEA2 algorithm. In following subsections, specific modifications done over the original algorithm are explained. These modifications can be summarized as follows:

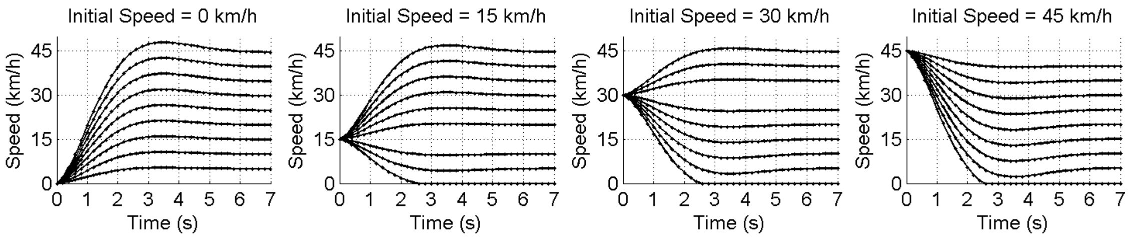

Figure 7: Evolution of the autonomous vehicle's speed to adjust to different references ($\{0, 5, 10, ...45km/h\}$) with different initial speeds ($\{0, 15, 30, 45\}km/h$).

- Individuals are executed in a randomly generated scenario, which changes every generation. This fact could make the fitness values obtained in one generation invalid for the next one, since a successful FRBS with success in one scenario may fail in the next one. This problem is solved by using a cumulative fitness value which evolves depending on both results obtained in previous and actual scenarios (Section 4.1).
- A method that dynamically changes the MFs used to codify an input variable has been developed. This is made with the aim of obtaining a small and meaningful set of input variables (Section 4.2).
- A biased initialization operator has been implemented. This operator is used to generate an initial Rule Base (RB) set with generality (Section 4.3).
- A crossover operator that uses information coming from the simulation used to evaluate individuals has been used. This operator identifies the most important rules in the both FRBS, in order to mix rules according to the importance in the whole controller behavior (Section 4.3).

4.1. Objective Functions

In this problem, we optimize FRBSs which guide the autonomous vehicle to a safe and efficient maneuvering. In this way, our first objective is to minimize the following time related measures, which define the performance and accuracy of the FRBS:

- Time to finish the maneuver (T_f): time needed by the autonomous vehicle to completely leave the *critical area*.
- Time in which two vehicles (the autonomous and other one) coincides inside the *critical area* (T_c): even if there is no collision between vehicles, two vehicles (the autonomous and one of the manuals) are not allowed to coincide in the critical area. Since the manual one pays no attention to actions from the other one, it is responsibility of the autonomous vehicle to avoid this situation.

Regarding these two measures, the objective of the proposed algorithm is to get a $T_c = 0$ and a minimum T_f. To simplify the number of objectives to optimize by the algorithm, both measures are joined in a unique one (O_{time}), calculated as follows:

$$O_{time} = 100 \cdot T_c + T_f \tag{14}$$

, as can be seen, this function highly penalizes controllers which produce collisions.

As commented before, the scenario used as benchmark in each generation is continuously changing. This fact allows the proposal to test FRBSs in many situations and it prevents the need of using a large amount of scenarios (with the computational cost that it would involve). These scenario's changes oblige the algorithm to re-evaluate individuals in each generation, in order to calculate objective functions obtained in the latter one.

To deal with this situation, once an individual is evaluated, the calculated O_{time} is weighted, making an average with the previous fitness value of the individual. This allows the algorithm to preserve information about FRBS's performance in previous scenarios, so the O_{time} measure is updated as follows:

$$O_{time} = 0.75 \cdot O_{time}(t-1) + 0.25 \cdot O_{time}(t) \tag{15}$$

where $(t-1)$ denotes cumulative values obtained previously, and (t) the current O_{time}.

This information preservation is inspired in similar techniques often used in signal processing for obtaining backgrounds in images or signals [34]. It is important to note that, after applying the crossover or mutation operator, an individual will preserve the O_{time} multiplied by a factor ($\vartheta > 1$), in order to slightly deteriorate the inherited fitness, which will be updated after its next evaluation.

Regarding the controller simplicity, we decide to minimize two of the most used complexity-based measures [13]. They define the length and complexity of the RBs:

- Number of rules (N_R), computed as the number of induced rules.
- Number of variables ($\overline{N_A}$), the number of variables of the antecedent. The number of variables for a rule set is computed as the average for each rule of that set.

They are also grouped as a single objective ($O_{simplicity}$), calculated as:

$$O_{simplicity} = N_R + \frac{\overline{N_A}}{N_V} \tag{16}$$

, where N_V represents the number of variables to be considered by the FRBS. $O_{simplicity}$ returns a real number, where the integer part denotes the number of rules, and the floating one the percentage of variables used in the antecedent. So, a FRBS with a small amount of complex rules is considered better than one with a large amount of simple ones.

4.2. Input Variables and Membership Functions

As commented in Section 3, the autonomous vehicle is able to access information related to speeds and times to reach and leave the critical area of all the involved vehicles (Figure 5). This information is received as speeds (s_x), distances to reach and to leave the critical area (rd_x and ld_x) and times to reach and to leave the critical area (rt_x and lt_x).

Since the number of variables is quite high (five variables for the autonomous vehicle and other five for each group of vehicles considered), in this work we propose a way to considerably simplify the number of variables involved in the scenario. We make it by using parametrized MFs which change in depending on the state of the vehicle.

Regarding the variables related to the state of the autonomous vehicle, two variables are used:

- $DIST_a$: This variable receives ld_a, but creates the MFs in function of both autonomous vehicle's distances (rd_a and ld_a).
- $TIME_a$; This variable receives lt_a, and creates the MFs according to rt_a and lt_a values.

The distribution of the MFs used to codify $DIST_a$ depends on the difference between rd_a and ld_a and an increment ψ_a^d. In the same way, MFs used to codify $TIME_a$ depends on the difference between rt_a and lt_a and ψ_a^t, as increment.

Variables referring to the autonomous vehicle are coded by four MFs, which denote whether the vehicle is $inside$, $entering$, $near$ and far from the critical area, both in distance or time. The $inside$ MFs are coded by a square function, since we consider that when the autonomous vehicle enters the critical area, it must take a constant action that guides it to finish the maneuver as soon as possible. MFs $entering$, $near$ and far are constructed by adding an increment ($\psi_a^{(d|t)}$) that will be set before the experimentation. In Figure 8 is shown how MFs are distributed in the space for $DIST_a$ (top) and $TIME_a$ (bottom).

In the same way as done with the autonomous vehicle, for each group of vehicles they are defined $DIST_i$ and $TIME_i$ to represent distances and times referring the $i-th$ group of vehicles.

The MFs used to codify these input variables are generated in a very similar way as done for the autonomous vehicle, with the difference that in case the vehicle is inside the critical area, two MFs are defined, representing if the vehicle just entered the critical area, or is leaving it ($Inside\ Entering$ and $Inside\ Leaving$). The distribution of MFs is done as presented in Figure 9, and, as has been done before, it depends on two increments ($\psi_i^{(d|t)}$) to be built.

To illustrate how MFs distribution changes depending on other variables, three examples are presented in Figure 10. In these examples, $lt_x = 30s$ (dashed line) for all of them, but the distribution of the MFs differs in each case, according to distance and speed values.

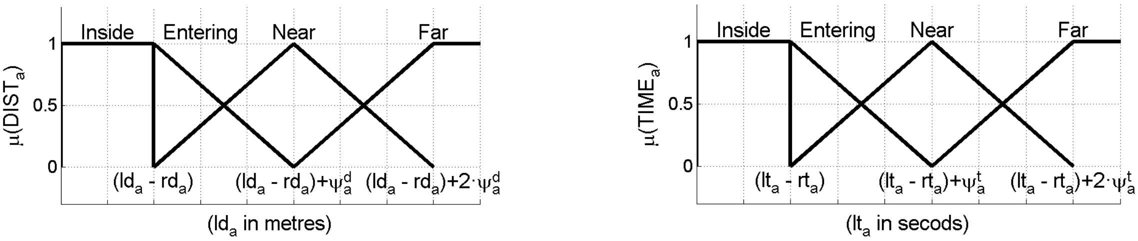

Figure 8: MF codification for variables $DIST_a$ (left) and $TIME_a$ (rigth).

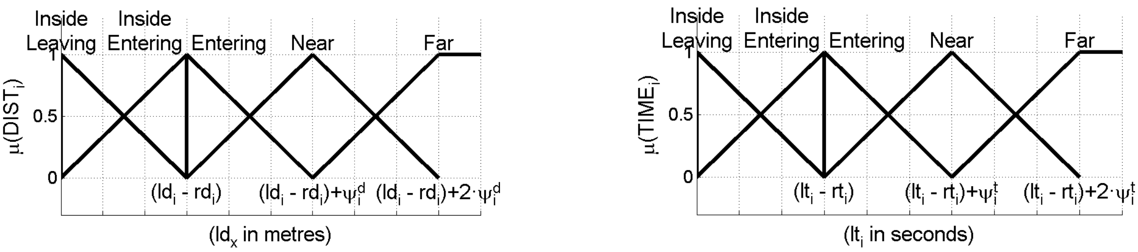

Figure 9: MF codification for variables $DIST_x$ (left) and $TIME_x$ (rigth).

4.3. Genetic Operators

In this section we introduce the initialization, crossover and mutation operators used in the developed algorithm. These operators are explained in detail next.

Codification: The solutions' codification is one of the most determining aspects in the characteristics of any evolutionary algorithm. The proposals in the literature follow different approaches to encode rules within a population of individuals [9]:

- The *Individual = Rule approach*, in which each individual codifies a single rule.
- The *Individual = Set of rules*, also called the Pittsburgh approach, in which each individual represents a set of rules.

In our proposal, we decide to use a Pittsburgh approach, since, for the proposed problem, it is hard to evaluate each rule in a separated way. This is because the FRBS behavior is determined by the cooperation of the whole rules set. We decide to use a matrix integer representation model, with so many rows as rules can be in the RB (Max_R), and as many columns as variables are used by the FRBS ($Input_N + 1$ including the output variable). Each row codifies a single rule with antecedent and consequent.

Each cell in the matrix $Ind(i, j)$ contains an integer in $[0, \overline{MF_j}]$ in the antecedent part and in $[1, \overline{MF_j}]$ for the consequent one, being $\overline{MF_j}$ the number of labels used to codify the $j - th$ variable. In Table 1 it is shown an example of RB codification. It is important to note that if a rule contains only zeros in the antecedent part, it does not count in the total number of rules of the FBRS.

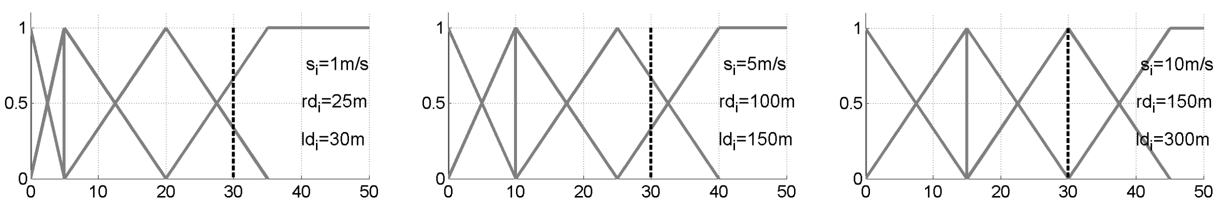

Figure 10: Illustrative example about how $TIME_x$ MFs are distributed depending on other state variables ($\psi_i^t = 15s$). Dashed line represents the lt_x used as input.

	V_1	V_2	Out	$Rule$
$Rule_1$	1	0	1	IF $\{V_1 = MF_{(1,V_1)}\}$ THEN $\{Out = MF_{(1,Out)}\}$
$Rule_2$	0	3	2	IF $\{V_2 = MF_{(3,V_2)}\}$ THEN $\{Out = MF_{(2,Out)}\}$
$Rule_3$	2	3	3	IF $\{V_1 = MF_{(2,V_1)}\}$ AND $\{V_2 = MF_{(3,V_2)}\}$ THEN $\{Out = MF_{(3,Out)}\}$
$Rule_4$	3	2	2	IF $\{V_1 = MF_{(3,V_1)}\}$ AND $\{V_2 = MF_{(2,V_2)}\}$ THEN $\{Out = MF_{(2,Out)}\}$

Table 1: Representation of a whole fuzzy RB with an integer matrix. In this case, $Max_R = 4$ and $Input_N = 2$.

Initialization: A biased initialization operator has been implemented [5]. The operator generates a percentage of the maximum number of rules (Max_R) using only a maximum percentage of variables for the antecedent ($Input_N$). Given two parameters (ψ_{ini} and ψ_{vars}), the initialization operator generates the ψ_{ini} (%) of the rules with only a maximum of ψ_{vars} (%) of the variables considered in the antecedent. Remaining rules ($1 - \psi_{ini}$) are generated in a fully random way. This operator aims to obtain an initial population of FRBSs with generic rules, which cover a high number of situations. In Algorithm 2 it is presented the pseudocode for the implemented initialization operator.

Data: N (population size)
Max_R (maximum number of rules per individual)
$Input_V$ (number of input variables of the problem)
ψ_{ini} (percentage of biased rules)
ψ_{vars} (percentage of variables used in biased rules)
Result: P_0 (Initial Population)
for $Ind = 1 \dots N$ **do**
  for $Rule = 1 \dots Max_R$ **do**
    if $rand() < psi_{ini}$ **then**
      for *Var=1...*$Input_V$ **do**
        if $rand() < psi_{vars}$ **then**
          $P_0(Ind, Rule, Var) = rand(\{0 \dots \overline{MF_{Var}}\})$
        else
          $P_0(Ind, Rule, Var) = 0$
        end
      end
    else
      for *Var=1...*$Input_V$ **do**
        $P_0(Ind, Rule, Var) = rand(\{0 \dots \overline{MF_{Var}}\})$
      end
    end
    $P_0(Ind, Rule, Input_V + 1) = rand(\{1 \dots \overline{MF_{Input_V+1}}\})$
  end
end

Algorithm 2: Pseudocode for the biased initialization used in this work.

Selection: is done by binary tournament [15], i.e., two individuals are randomly chosen. Then, their fitness is compared, and the one with the greater value is selected as parent.

Crossover: Uniform crossover [10] is used to obtain offsprings from selected parents. The operator makes a random choice as to which parent it should each gene be inherited from. This is implemented by generating a string of random variables from a uniform distribution in [0, 1]. For each position, if the value is below 0.5, the gene is inherited from the first parent, and otherwise from the second. The second offspring is created using the inverse string [39].

The novelty implemented in this work is the sorting applied to rules in both parents before applying the operator. Rules are sorted by their cumulative activation during the last simulation. With this rule order, we assure that most and least activated rules in parents are equally distributed among offsprings. In Figure 11, the effect of the sorting over

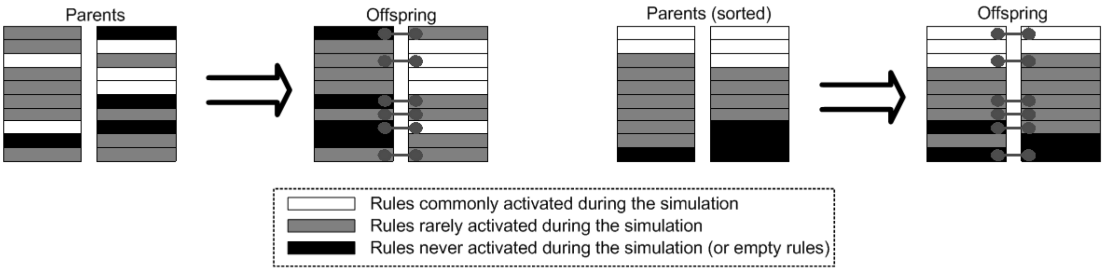

Figure 11: Example of application of the uniform crossover over two activation-sorted RBs. Resulting offspring without sorting the parent's RB (left) and sorting them (rigth).

the offspring can be seen in a graphical way. As commented before, offsprings' O_{time} are degraded with a factor (ϑ) after applying crossover; due to this reason, it is used 0.75 (instead of 0.5) as threshold to decide from which parent an offspring receives a gene.

Mutation: The mutation implementation is done by a random variation method: for each gene, the current value is increased or decreased by 1 respecting the set of permissible values (in $[0, \ldots, \overline{MF_i}]$ for antecedent genes and $[1, \ldots, \overline{MF_i}]$ for consequent ones) with probability $pm = 1/N_R$.

5. Experimentation

To evaluate the proposed method, the problem presented in Section 3 has been considered for resolution. This section is organized as follows: Section 5.1 presents the experimental setup, Section 5.2 analyzes the obtained results, finally Section 5.3 shows a qualitative evaluation of some selected FRBS, analyzing differences among them.

5.1. Experimental Setup

The proposal has been run using the parameters detailed below:

- Regarding the SPEA2 algorithm(Section 2.1), the population and archive size is 50, and the generations number is 500.
- Scenarios (Section 3.1) are initialized with the autonomous vehicle starting at 100m from the intersection, and at 50km/h speed. Vehicle's length is established at 4m. Regarding the manual vehicles, ten of them are randomly generated with a random length between [5,20]m and inter spaces between them in [10,40]m. All the manual vehicles circulate at the same speed, generated in [10,30]km/h. In addition, the time interval over which two vehicles are merged as a group (T_{inter}) is established at 2.5s.
- Factor in which an offspring is degraded in relation with the fitness accumulated by its parent is established at 1.25 (ϑ in Section 4.1).
- The FRBSs use six input variables. They are the pairs ($TIME$ and $DIST$ in Section 4.2) related, respectively to the autonomous vehicle, the first and the second group of vehicles: $\{TIME_{\{a|1|2\}}, DIST_{\{a|1|2\}}\}$.
- Increments over the time and distance differences used to define the MF distribution (in Section 4.2) are set as $\psi_a^d = 25m$ and $\psi_a^t = 5s$ for the autonomous vehicle, and $\psi_i^d = 10m$ and $\psi_i^t = 5s$ for manual ones.
- Finally, in the initialization operator, $\psi_{ini} = 0.75$ and $\psi_{vars} = 0.25$ (see Section 4.3) parameters are used. In other words, the 75% of the rules are initialized with only the 25% of the input variables ($Input_N = 6$). In total, individuals are initialized with $Max_R = 50$ rules at most.

5.2. Results analysis

This section analyzes the results obtained by the proposed algorithm over ten runs and with parameters described in previous section. The proposal was coded in MATLAB[1] and ran in a Intel Core i5 2410 laptop, with 2.30 GHz and

[1]http://www.mathworks.es/products/matlab/

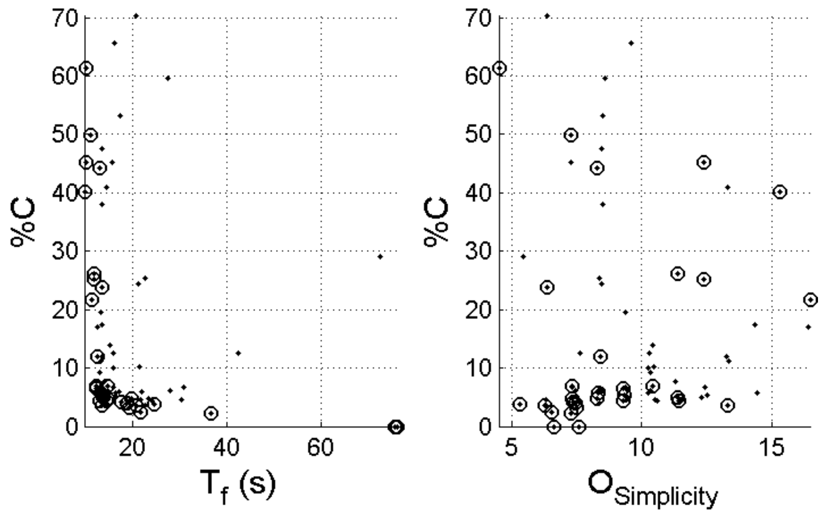

Figure 12: Individuals in the 10 obtained paretos evaluated in terms of $\{\%C, \overline{T_f}, O_{Simplicity}\}$. Marked dots represent the non-dominated individuals in terms of the mentioned objectives.

a RAM of 4 GB. The computational time needed for each run is strictly ligated with the time needed to simulate the execution of the FRBS in the environment. In particular, each run last for near 90 minutes, of which more than 95% is dedicated to run simulations.

Note that O_{time} is not suitable to compare results between different runs; this is because where all the FRBSs fail, will increase the O_{time} value in all the individuals in the population, as well as a quite scenario will make all the O_{time} values to decrease. So the final O_{time} value is highly dependent of the sequence of generated scenarios.

Due to that, all the individuals in the pareto sets from each run were subjected to an exhaustive experimentation, where individuals were evaluated in 500 random scenarios. Over this evaluation, the following performance measures were taken:

- Percentage of scenarios where the FRBS produces a coincidence between the autonomous and a manual vehicle in the critical area ($\%C$).
- Averaged time to cross the intersection ($\overline{T_f}$ in Section 4.1).
- Controller simplicity ($O_{Simplicity}$ in Section 4.1).

With these three measures, Figure 12 shows the results of all the individuals in the ten obtained paretos. In Table 2 results of non-dominated controllers are summarized , where the $O_{Simplicity}$ measure has been decomposed in the total number of rules (N_R) and average number of variables per rule ($\overline{N_V}$) for a better understanding.

Analyzing the table, it can be seen that most of non-dominated FRBSs have a $\%C$ < 10% (it is important to note that not all the coincidences imply a collision between both vehicles).

Since the autonomous vehicle started always at the same distance from the intersection point (100m), the time to finish (T_f) gives us an idea about how fast the FRBS drive. In the table we find FRBSs that never produce coincidences but with high T_f (see $FRBS_1$ and $FRBS_2$). These cases define too cautious controllers, or controllers which stop waiting for a very big inter space between manual vehicles to cross.

In the other hand, we got controllers with a high $\%C$ but with very short T_f (see $FRBS_{25}$ to $FRBS_{30}$); which are controllers that almost do not pay attention to the incoming traffic, neither reducing their speed. In the extreme we have the controller with highest $\%C$, which is also the simplest one, with only four rules ($FRBS_{30}$).

In general terms, controllers simplicity is well distributed. N_R varies from 4 to 16, being about 8 the averaged number of controllers rules, while N_V does the same between 1.6 and 3.8.

In order to complete a deeper analysis of the FRBS behavior, in next subsection, some controllers are selected and their behaviors will be analyzed in a particular scenario.

Individual	%C	$\overline{T_f}$	N_R	$\overline{N_V}$
$FRBS_1$	**0.0**	75.9	**6**	3.8
$FRBS_2$	**0.0**	75.6	**7**	3.6
$FRBS_3$	**2.2**	36.4	**7**	**1.6**
$FRBS_4$	**2.4**	21.5	**6**	3.2
$FRBS_5$	**3.2**	19.1	**7**	2.9
$FRBS_6$	**3.6**	13.4	13	1.9
$FRBS_7$	**3.6**	20.6	**6**	**1.7**
$FRBS_8$	3.8	24.5	**5**	**1.8**
$FRBS_9$	4.0	18.8	**7**	2.7
$FRBS_{10}$	4.2	17.6	**7**	2.3
$FRBS_{11}$	4.4	14.3	9	**1.6**
$FRBS_{12}$	4.4	12.9	11	2.5
$FRBS_{13}$	4.8	14.6	8	**1.8**
$FRBS_{14}$	4.8	19.7	**7**	1.9
$FRBS_{15}$	5.0	13.7	11	2.2
$FRBS_{16}$	5.4	13.4	9	2.0
$FRBS_{17}$	5.8	13.2	8	2.0
$FRBS_{18}$	6.6	12.3	9	**1.8**
$FRBS_{19}$	7.0	14.6	**7**	2.0
$FRBS_{20}$	7.0	12.1	10	2.5
$FRBS_{21}$	12.0	12.4	8	2.6
$FRBS_{22}$	21.6	**11.3**	16	2.9
$FRBS_{23}$	23.8	13.4	**6**	2.2
$FRBS_{24}$	25.2	**11.8**	12	2.3
$FRBS_{25}$	26.2	**11.6**	11	2.2
$FRBS_{26}$	40.2	**9.9**	15	2.0
$FRBS_{27}$	44.2	12.8	8	**1.8**
$FRBS_{28}$	45.2	**10.0**	12	2.3
$FRBS_{29}$	49.8	**10.9**	**7**	**1.6**
$FRBS_{30}$	61.4	**10.0**	**4**	3.0

Table 2: Results of non-dominated individuals in evaluation over 500 scenarios. Sorted by $\%C$. For each column, the 25% minimum have been bolded for a better understanding.

5.3. Qualitative Analysis

This section presents a qualitative evaluation of some of the controllers in simulations with the purpose of discussing the actions taken to control the vehicle.

We select the individuals $\{FRBS_3,\ FRBS_6,\ FRBS_7\}$ from Table 2 to perform a more qualitative analysis. Their RBs are presented in Table 3. These three individuals have been selected since all of them have $\%C < 5\%$. In addition, $FRBS_6$ performs the best $\overline{T_f}$ value for all individuals with $\%C < 5\%$, and it is one of the most complex one (13 rules). $FRBS_3$ has the second less $\%C$ value with a reasonable simplicity and $\overline{T_f}$. Finally, $FRBS_7$ is even more simple and fast than $FRBS_3$ without an excessive increase in $\%C$.

From a first view over the obtained RBs, the rules simplicity can be seen, having most of them only one or two variables involved in the antecedent. On the other hand, some minor inconsistencies can be observed between pairs of rules $\{R_{(3,6)} - R_{(3,7)}\}$; $\{R_{(6,1)} - R_{(6,2)}\}$ and $\{R_{(6,10)} - R_{(6,11)}\}$; whose have the same antecedent but different consequent. This inconsistencies can be solved, and RBs could be simplified by using the same antecedent and, as consequent, the average value in the two rules.

For the qualitative comparison, the same randomized scenario has been selected for the three FRBSs. Figures 13, 14 and 15 show the behavior carried out by $FRBS_3$, $FRBS_6$ and $FRBS_7$, respectively. Figures show the autonomous vehicle's speed under the FRBS guidance (top) and the remaining distance and time to reach and leave

Rule	$DIST_a$	$TIME_a$	$DIST_1$	$TIME_1$	$DIST_2$	$TIME_2$	Out
$R_{(3,1)}$	-	-	-	-	$Inside_{Leaving}$	-	20
$R_{(3,2)}$	-	-	-	Far	-	-	10
$R_{(3,3)}$	-	-	$Inside_{Entering}$	-	$Entering$	-	40
$R_{(3,4)}$	-	Far	-	-	-	-	50
$R_{(3,5)}$	$Entering$	-	$Inside_{Entering}$	Far	$Near$	-	30
$R_{(3,6)}$	$Near$	-	-	-	-	-	40
$R_{(3,7)}$	$Near$	-	-	-	-	-	50

Rule	$DIST_a$	$TIME_a$	$DIST_1$	$TIME_1$	$DIST_2$	$TIME_2$	Out
$R_{(6,1)}$	-	-	-	-	-	$Inside_{Entering}$	20
$R_{(6,2)}$	-	-	-	-	-	$Inside_{Entering}$	40
$R_{(6,3)}$	-	-	-	-	$Inside_{Entering}$	-	40
$R_{(6,4)}$	-	-	$Near$	Far	-	-	30
$R_{(6,5)}$	-	-	Far	-	-	-	50
$R_{(6,6)}$	-	-	Far	Far	-	$Inside_{Entering}$	50
$R_{(6,7)}$	-	$Near$	-	-	-	-	10
$R_{(6,8)}$	$Near$	-	-	-	-	-	30
$R_{(6,9)}$	$Near$	-	$Inside_{Entering}$	$Inside_{Entering}$	$Inside_{Leaving}$	$Entering$	40
$R_{(6,10)}$	Far	-	-	-	-	-	10
$R_{(6,11)}$	Far	-	-	-	-	-	50
$R_{(6,12)}$	Far	-	$Entering$	Far	$Inside_{Leaving}$	$Entering$	50
$R_{(6,13)}$	Far	$Entering$	-	-	-	-	10

Rule	$DIST_a$	$TIME_a$	$DIST_1$	$TIME_1$	$DIST_2$	$TIME_2$	Out
$R_{(7,1)}$	-	-	-	-	$Near$	-	20
$R_{(7,2)}$	-	-	Far	-	-	-	30
$R_{(7,3)}$	-	-	Far	-	-	-	40
$R_{(7,4)}$	-	Far	-	-	-	-	20
$R_{(7,5)}$	$Inside$	-	-	Far	$Inside_{Entering}$	$Inside_{Leaving}$	30
$R_{(7,6)}$	$Near$	-	-	-	$Inside_{Entering}$	-	40

Table 3: $FRBS_3$ Rule base (top) $FRBS_6$ Rule base (center) and $FRBS_7$ Rule base (bottom). (-) means *any* or variable not used.

the critical area (center and bottom). In graphics referring to distance and time, the ones concerning the first vehicle (or group) have been presented in red, and in blue for the second group. Note that, once one of the manual vehicles leaves the critical area, values for the first and second vehicles group change. Finally, the gray area shows values referring to the autonomous vehicle.

Next we proceed to remark some observations done over figures:

- $FRBS_3$ (Figure 13) starts decreasing the speed. After that, it is increased (seconds 5 to 10), and then, the vehicle gets stop near the critical area, waiting for a safe inter space in order to proceeds with the maneuver (before second 25). It finalizes the maneuver after 25.4s, being the slowest (and most cautious) one, as concluded from Table 2.

- $FRBS_6$ (Figure 14) starts maintaining the maximum speed of 50km/h; after the fourth second, it decreases the speed (until $\approx 10km/h$) to let the vehicle inside the critical area to leave it. Then it proceed to accelerate and cross the intersection before the following vehicle enters. This controller is the faster one, finalizing the maneuver in only 10.2s.

- $FRBS_7$ (Figure 15) starts with a similar behavior to the one shown by $FRBS_3$ but, after second 10, instead of stopping the vehicle, it maintains a low speed ($\approx 10km/h$) until it gets the opportunity of crossing, finalizing the maneuver in 18s.

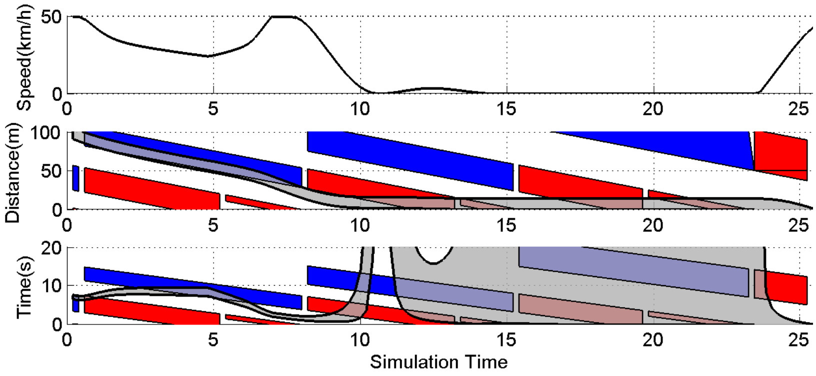

Figure 13: Behavior of $FRBS_3$ over the test scenario.

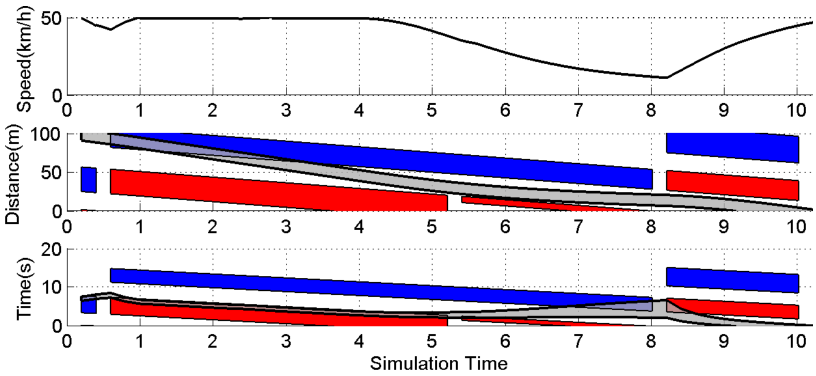

Figure 14: Behavior of $FRBS_6$ over the test scenario.

Results obtained confirm the good performance of the obtained FRBSs, as well as the diversity of behaviors obtained.

6. Conclusions and Future Works

In this work, an evolutionary multi-objective model for the induction of simple fuzzy rules is presented. It describes the speed profile to implement by an autonomous vehicle which approaches to an intersection crossed by a row of vehicles.

Manual vehicles approaching to the intersection are supposed to do not pay attention to the autonomous one, so, a coordination scheme is not valid in this situation. With this scenario, it is necessary for the autonomous vehicle to execute a decision process which advise it about which is the proper speed to circulate along the intersection.

To achieve this goal, a FRBS structure is defined, where MFs defining distances and times to reach/leave the intersection vary over time and in dependance with the state of the traffic. The set of rules that define the behavior of the FRBS is coded and evolved by a SPEA2 based process, which optimizes four aspects of the controller: (i) accuracy (less coincidence in intersection with other vehicles), (ii) performance (minimum time to finish the maneuver) (iii) length of the RB (less number of rules) and (iv) complexity of the RB (less number of antecedents). These four measures are merged into only two, summarizing accuracy-performance of the maneuvering by one, and length-complexity by the other one.

Since the evaluations of the individuals require simulation executions, it has been implemented a method that makes that the measure referring to accuracy-performance can be *inheritable* among generations. With this, the individuals in one generation need only to be evaluated in one scenario and, considering this *inheritable* value, establish a general fitness measure that summarizes the quality of the individual over all the scenarios used in previous genera-

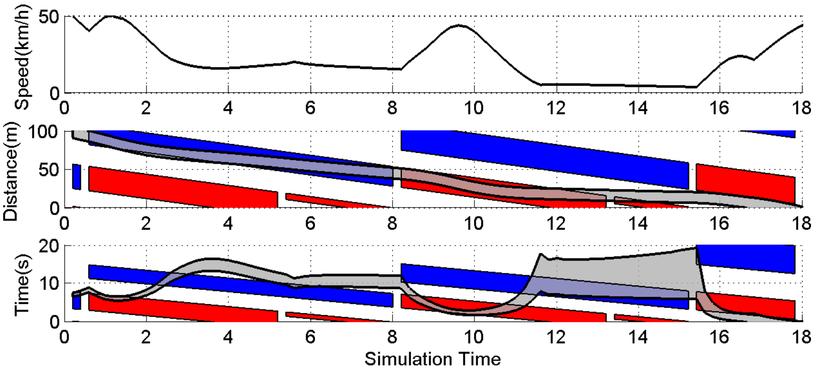

Figure 15: Behavior of $FRBS_7$ over the test scenario.

tions. This avoids the evolutionary process to execute a long simulation (or a large number of simulations) to evaluate the FRBSs performance in a large set of situations.

In addition, genetic operators have been modified with the purpose of adapting them to the specific problem, by using a special initialization operator. In this way, general and simple FRBS are obtained in the first generation, while by ordering the rules used before crossing individuals, controllers are crossed in a more balanced way.

The optimization process was executed 10 times, and all the paretos were merged to obtain the non-dominated from the non-dominated individuals in each run. They were tested under an exhaustive set of 500 scenarios, proving the solutions ability to satisfactory deal with a large set of situations. In addition, a noticeable simplicity is observed over the obtained FRBSs.

Future works will be in the line of using the presented methodology to more complex traffic situations, such overtaking in two way roads. Moreover, it is expected to use the proposal in situations where information about a larger number of vehicles is required. Finally, the present system will be coordinated with others that are used in autonomous vehicles, such as adaptive cruise control depending on a leading vehicle.

ACKNOWLEDGMENT

The authors would like to thank the EU *Intelligent Cooperative Sensing for Improved traffic efficiency* (ICSI) project (FP7-ICT-2011-8) for its support in the development of this work.

References

[1] Aström, K., Hägglund, T., 2001. The future of PID control. Control Engineering Practice 9 (11), 1163 – 1175.

[2] Badie, A., 2010. Genetic fuzzy self-tuning PID controllers for antilock braking systems. Engineering Applications of Artificial Intelligence 23 (7), 1041–1052.

[3] Baker, C., Dolan, J., 2009. Street smarts for Boss: Behavioral subsystem engineering for the Urban Challenge. IEEE Robotics and Automation Magazine 16 (1), 78–87.

[4] Bishop, R., 2000. Survey of intelligent vehicle applications worldwide. In: Proceedings IEEE Intelligent Vehicles Symposium. pp. 25–30.

[5] Carmona, C., González, P., Jesús, M., Herrera, F., 2010. NMEEF-SD: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. IEEE Transactions on Fuzzy Systems 18 (5), 958–970.

[6] Casillas, J., Cordon, O., Del Jesus, M., Herrera, F., 2001. Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems. Information Sciences 136 (1-4), 135–157.

[7] Casillas, J., Herrera, F., Pérez, R., del Jesus, M., Villar, P., 2007. Special issue on genetic fuzzy systems and the interpretability-accuracy trade-off. International Journal of Approximate Reasoning 44 (1), 1–3.

[8] Coello, C., Lamont, G., Van-Veldhuizen, D., 2007. Evolutionary algorithms for solving multi-objective problems. Springer.

[9] Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L., 2001. Recent advances in genetic fuzzy systems. Information Sciences 136 (1-4), 1–5.

[10] De Jong, K. A., 1975. An analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, University of Michigan.

[11] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6 (2), 182–197.

[12] Dion, F., Hellinga, B., 2002. A rule-based real-time traffic responsive signal control system with transit priority: Application to an isolated intersection. Transportation Research Part B: Methodological 36 (4), 325–343.

[13] Fazzolari, M., Alcalá, R., Nojima, Y., Ishibuchi, H., Herrera, F., 2013. A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions. IEEE Transactions on Fuzzy Systems 21 (1), 45–65.

[14] Godoy, J., Milanés, V., Pérez, J., Villagrá, J., Onieva, E., 2013. An auxiliary V2I network for road transport and dynamic environments. Transportation Research Part C: Emerging Technologies 37, 145–156.

[15] Goldberg, D. E., Deb, K., 1991. A comparative analysis of selection schemes used in genetic algorithms. In: Foundations of Genetic Algorithms. Morgan Kaufmann, pp. 69–93.

[16] Goldberg, D. E., Holland, J. H., 1988. Genetic algorithms and machine learning. Machine Learning 3 (2), 95–99.

[17] Hafner, M., Cunningham, D., Caminiti, L., Del Vecchio, D., 2013. Cooperative collision avoidance at intersections: Algorithms and experiments. IEEE Transactions on Intelligent Transportation Systems PP (99), 1–14.

[18] Holland, J. H., 1992. Adaptation in natural and artificial systems. MIT Press, Cambridge.

[19] Ioannou, P., Chien, C., 1993. Autonomous intelligent cruise control. IEEE Transactions on Vehicular Technology 42 (4), 657–672.

[20] Kammel, S., Ziegler, J., Pitzer, B., Werling, M., Gindele, T., Jagzent, D., Schröder, J., Thuy, M., Goebl, M., von Hundelshausen, F., Pink, O., Frese, C., Stiller, C., 2008. Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge. Journal of Field Robotics 25 (9), 615–639.

[21] Konar, A., 2005. Computational intelligence: principles, techniques and applications. Springer.

[22] Lee, J.-H., Lee-Kwang, H., 1999. Distributed and cooperative fuzzy controllers for traffic intersections group. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 29 (2), 263–271.

[23] Lefevre, S., Laugier, C., Ibanez-Guzman, J., 2012. Evaluating risk at road intersections by detecting conflicting intentions. In: IEEE International Conference on Intelligent Robots and Systems. pp. 4841–4846.

[24] Li, L., Wang, F.-Y., 2006. Cooperative driving at blind crossings using intervehicle communication. IEEE Transactions on Vehicular Technology 55 (6), 1712–1724.

[25] Li, Y., Fan, X., 2003. Design of signal controllers for urban intersections based on fuzzy logic and weightings. In: Proceedings of IEEE Intelligent Transportation Systems. Vol. 1. pp. 867–871.

[26] Mamdani, E., 1974. Application of fuzzy algorithms for control of simple dynamic plant. Proceedings of the Institution of Electrical Engineers 121 (12), 1585–1588.

[27] Milanés, V., Pérez, J., Onieva, E., González, C., 2010. Controller for urban intersections based on wireless communications and fuzzy logic. IEEE Transactions on Intelligent Transportation Systems 11 (1), 243–248.

[28] Milanés, V., Villagrá, J., Godoy, J., Simó, J., Pérez, J., Onieva, E., 2012. An intelligent V2I-based traffic management system. IEEE Transactions on Intelligent Transportation Systems 13 (1), 49–58.

[29] Naranjo, J. E., Gonzalez, C., Garcia, R., de Pedro, T., Sotelo, M. A., 2007. Using fuzzy logic in automated vehicle control. IEEE Intelligent Systems 22 (1), 36–45.

[30] Oh, S., Jang, H.-J., Pedrycz, W., 2011. A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization. Expert Systems with Applications 38 (9), 11217–11229.

[31] Onieva, E., Milanés, V., González, C., De Pedro, T., Pérez, J., Alonso, J., 2010. Throttle and brake pedals automation for populated areas. Robotica 28 (4), 509–516.

[32] Papadimitratos, P., La Fortelle, A., Evenssen, K., Brignolo, R., Cosenza, S., 2009. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. IEEE Communications Magazine 47 (11), 84–95.

[33] Piao, J., McDonald, M., 2008. Advanced driver assistance systems from autonomous to cooperative approach. Transport Reviews 28 (5), 659–684.

[34] Radke, R., Andra, S., Al-Kofahi, O., Roysam, B., 2005. Image change detection algorithms: A systematic survey. IEEE Transactions on Image Processing 14 (3), 294–307.

[35] Röglinger, S., 2011. A methodology for testing intersection related Vehicle-2-X applications. Computer Networks 55 (14), 3154–3168.

[36] Spek, A., Wieringa, P., Janssen, W., 2006. Intersection approach speed and accident probability. Transportation Research Part F: Traffic Psychology and Behaviour 9 (2), 155–171.

[37] Staubach, M., 2009. Factors correlated with traffic accidents as a basis for evaluating advanced driver assistance systems. Accident Analysis & Prevention 41 (5), 1025–1033.

[38] Sugeno, M., 1999. On stability of fuzzy systems expressed by fuzzy rules with singleton consequents. IEEE Transactions on Fuzzy Systems 7 (2), 201–224.

[39] Sywerda, G., 1989. Uniform crossover in genetic algorithms. In: Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 2–9.

[40] Tari, T., Koczy, L., Gaspar, C., Hontvari, J., 2006. Control of traffic lights in high complexity intersections using hierarchical interpolative fuzzy methods. In: IEEE International Conference on Fuzzy Systems. pp. 1045–1048.

[41] Tejado, I., Milanés, V., Villagrá, J., Godoy, J., HosseinNia, H., Vinagre, B., 2011. Low speed control of an autonomous vehicle by using a fractional pi controller. pp. 15025–15030.

[42] Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.-W., Singh, S., Snider, J., Stentz, A., Whittaker, W., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., Ferguson, D., 2008. Autonomous driving in urban environments: Boss and the Urban Challenge. Journal of Field Robotics 25 (8), 425–466.

[43] Vanderhaegen, F., 2012. Cooperation and learning to increase the autonomy of ADAS. Cognition, Technology and Work 14 (1), 61–69.

[44] Wang, J., Xu, W., Gong, Y., 2010. Real-time driving danger-level prediction. Engineering Applications of Artificial Intelligence 23 (8), 1247–1254.

[45] Wu, Y., Zhang, B., Lu, J., Du, K., 2011. Fuzzy logic and neuro-fuzzy systems: A systematic introduction. International Journal of Artificial Intelligence and Expert Systems 2 (2), 47–80, cited By (since 1996) 5.

[46] Yagar, S., Han, B., Greenough, J., 1992. Real-time signal control for mixed traffic and transit based on priority rules. Traffic Management,

Engineering Foundation Press, 147–166.
[47] Zadeh, L., 1965. Fuzzy sets. Information and Control 8 (3), 338–353.
[48] Zitzler, E., Laumanns, M., Thiele, L., 2001. SPEA2: Improving the strength pareto evolutionary algorithm. Tech. rep., Swiss Federal Institute of Technology, Lausanne, Switzerland.
[49] Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3 (4), 257–271.