A constructivist approach for a self-adaptive decision-making system: application to road traffic control

Maxime Guériau*,†, Fréderic Armetta*, Salima Hassas*, Romain Billot‡ and Nour-Eddin El Faouzi†
*Univ. Lyon, UMR CNRS 5205 LIRIS, F-69622 Villeurbanne, France
Email: {maxime.gueriau, frederic.armetta, salima.hassas}@liris.cnrs.fr
†Traffic Engineering Laboratory (LICIT), Univ. Lyon, F-69000 Lyon, France
IFSTTAR, LICIT, F-69675 Bron – ENTPE, LICIT, F-69518 Vaulx En Velin, France
Email: {maxime.gueriau, nour-eddin.elfaouzi}@ifsttar.fr
‡Institut Mines-Télécom, Télécom Bretagne, UMR CNRS 6285 Lab-STICC
Technopôle Brest Iroise, CS 83818, 29238 Brest Cedex 3, France
Email: romain.billot@telecom-bretagne.eu

***Abstract*—The relevance of decision making in autonomous systems is intrinsically related to the system capacity to discriminate its perception-action states. This is particularly challenging in unknown and changing complex environments, where providing a complete *a priori* representation to the system is not possible. To illustrate the problem, let us consider a decentralized control of road traffic, where a control device of the distributed infrastructure locally controls traffic, by learning to construct a precise representation (perception-action states) of the traffic state. In this context, it is challenging to define from prior knowledge a relevant representation of the traffic state that enables an efficient recommendation-based control. Without considering a prior domain-knowledge representation, we propose an approach able to combine a set of existing traditional unsupervised learning methods that collaborate as a population of agents in order to build an efficient representation. Our approach follows a constructivist learning perspective, where each agent produces a possible discretization of the raw sensed data. Thanks to a multi-agent reinforcement learning process, the population is able to collectively build a representation that combines the good capacities of the individual ones.**

I. Problem Statement

A. Context and challenges

The relevance of autonomous decision-making systems lays in their capacity to discriminate their perception-action states. This capacity strongly depends on the accuracy of the representation that the system has on its environment and on the decision-making problem to address. Traditional AI approaches provide these representations a priori, as part of domain-specific knowledge. However, defining an accurate representation of an *a priori* unknown and dynamic environment is a hard task. The challenging issue is thus, how to make an autonomous decision-making system able to construct a representation that allows accurate decision-making. To do so, we consider the perspective of constructivist learning, where the system builds a mapping of its environment into a set of perception-action states and the knowledge that governs the decision process, through learning from its experience of interaction with its environment.

As an illustration of a complex decision making environment, let us consider the case study of road traffic control in the context of Cooperative Intelligent Transportation Systems (C-ITS), as shown in Figure 1.

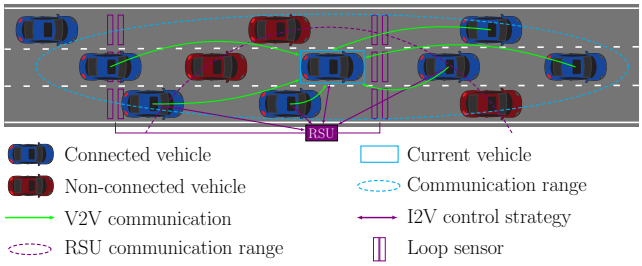


Fig. 1: General framework of Cooperative Intelligent Transportation Systems

These complex systems are based on recent advances in communication and information technologies to efficiently improve traffic flow. In a near future, it is expected that the road infrastructure will be shared by connected and non-connected vehicles. Connected vehicles will make use of wireless communication to exchange information with other vehicles (V2V – vehicle to vehicle communication) or with the infrastructure (V2I – vehicle to infrastructure; I2V infrastructure to vehicle). In C-ITS, intelligent infrastructure controllers, also known as Road Side Units (RSU), are in charge of collecting information on their dedicated section. More than a simple relay of decentralized information, a RSU may become a traffic road controller able to disseminate recommendations to connected vehicles through I2V communication.

In this context, a challenging objective would be the use of Road Side Units to locally and dynamically control the traffic flow. The controller system will have to tackle a high-

dimensional data space in this environment (percentage of connected vehicles, drivers behavior variability, road network topology, etc.) and to develop its capacity to rapidly self-adapt to new situations. For this purpose, we propose a constructivist learning approach that allows to ground the system knowledge in its current experience, through interaction with the considered environment.

B. Contributions and organization of the paper

The first objective of this work is to propose a generic model of an autonomous system with the ability to build a representation from its environmental interactions considered at their lowest level (Section II). The model follows a constructivist perspective. It couples two co-evolving processes: the construction of "perception-action states" hypothesis and the elaboration of the "control strategy". These processes are handled by a population of competing agents to construct an effective representation of the environment (set of "perception-action" states) and a decision mechanism allowing the selection of an accurate action. This last mechanism is seen as a reinforcement learning process where feedback is provided by the environment aiming to reinforce the perception-action state hypothesis proposed by the agents. The originality of our contribution relies on the use of concurrent representations that are supported by autonomous agents and that are adapted through collective intelligence mechanisms, to produce an emergent representation, in a dynamic context.

The second objective of the paper is to come up with an innovative application of our conceptual model. We choose the case study of connected traffic control that represents a complex environment where a controller agent implements our model. The controller agent is a Road Side Unit controlling a road section containing connected vehicles. As described in Section III, experiments show that even a simple implementation of the model leads to an improvement of the traffic flow. Nevertheless, we also identify several ways to improve the model when discussing the results.

C. Related Work

Traditional Artificial Intelligence (AI) approaches rely on an abstraction of the environment proposed by a system designer. This representation strongly depends on the problem specificities and the available environment (sensors/ actuators). Its adaptation to different problems/environments is thus limited [1]. As one of the main characteristics of autonomy is the capacity of adaptation, constructivist approaches suggest to make an autonomous agent iteratively construct a representation of its environment, based on its experience of interaction with it. These approaches are inspired by the seminal work of Piaget [2] in cognitive science, on the child cognitive development, and leads to the definition of principles to follow in designing autonomy [3]: embodiment, situatedness and a prolonged epigenetic developmental process.

Epigenetic has been introduced by Piaget and refers to the individual development through interactions with its environment. In such a defined problem, according to [4], there is a need to work with reasonably realistic physical worlds which do not oversimplify the learner's experience. The most promising approach to developmental AI is to reproduce the path along which humans acquire knowledge. A seminal attempt tried to construct a representation from sensori-motor interactions [5]. These approaches face a fundamental issue known as the bootstrap problem, related to the difficulty to initiate the world representation construction, without any prior (domain-specific) knowledge.

Some constructivist approaches propose to provide the system with a few low-level mechanisms in order to bootstrap the learning process [6], [7]. The authors' model is based on the use of Growing Neural Gaz (GNG) [8] - an adaptation of the Self-Organizing Maps (SOM) [9] - to provide the system with a first discretization of the available sensory experience. These approaches allow to reduce the dimensions of the perceived data (the input space), to simplify the identification of relevant actions in such a restricted space (the set of relevant actions represent a policy). One can note that this discretization results from a prior partitioning of the perceived data, and does not fully embrace the framework defined by constructivism that would consider it as the result of interaction with the environment. Indeed, such a pre-established discretization can appear inappropriate, but the system can make it evolve. One way to tackle this problem is to make the system use different discretizations, and combine them to adopt the best ones.

In this paper, we propose to go a step further by making the system consider several concurrent representations, that are updated in parallel. These abstractions (interpretations) of the raw sensed data are generated off-line, from different clustering algorithms (K-means [10], for instance), or on-line (using the same GNG algorithm). To enable this kind of concurrent processes and their interactions, we use the multi-agent systems (MAS) paradigm. The model we present here, offers generic mechanisms to help an autonomous control system in the construction of its representation of the environment through concurrent learning processes. It couples two co-evolving processes: the construction of "perception-action states" and the elaboration of the "control strategy". These processes are handled by a population of competing agents to construct an effective representation of the environment (set of "perception-action" states) and a decision mechanism allowing the selection of an accurate action. This last mechanism is seen as a reinforcement learning problem, where feedback is provided by the environment to reinforce the perception-action state hypotheses proposed by the agents.

From the traffic engineering standpoint, traffic control relies on a set of expert rules to be applied in a predefined discrete context. These strategies can be either applied at a system level [11] or at a local scale [12], [13]. Agent-based technologies have also been widely applied to deal with the control and the management of transportation systems [14]. Most of the previous contributions focus on traffic lights coordination and other forms of intersections [15], [16]. However, especially due to connected vehicles, we can observe a recent interest on highway traffic control. Such approaches (as proposed in [17])

exploit dynamic and connected strategies, such as Variable Speed Limits, to propose more distributed control strategies.

II. Building a representation from concurrent learning

The starting point in our model is the design of a dynamic decision-making system able to clearly define the boundaries of each perception-action state, in order to accurately identify the action perimeter while considering a set of perceptions. Indeed, we can expect from such a system a better understanding of the dynamics of the environment and a greater ability to select optimal actions according to the estimated context. Then, several challenges have to be faced in order to design a suitable model. First of all, one has to select the raw material provided to the system without incorporating too much domain-specific knowledge. Second, the ideal perimeter of the actions is not reachable in many cases. The issue is to find the best way to model these boundaries without limiting the set of reachable states. Finally, the most interesting challenge is to reinforce representations leading to an efficient control of the environment. We propose a generic conceptual model, with the general mechanisms matching the previous challenges, that could be easily adapted in their implementation to fit the problem requirements.

A. Overview

The proposed model relies on internal collective intelligence to assist the system in the iterative construction of its inner representation, described as a coupling between a perception and decision processes. The model can be described as a traditional perception-decision process of a controller system, as illustrated in Figure 2. Inputs, gathered by sensors, are observations of the environment composing the low-level perception of the system. These inputs are continuous and unlabeled, since no expert information is provided to the system, neither about the relevance nor about the importance or the given data. The input data is then assigned to internal "discretizer" agents in charge of providing abstractions of the transmitted information. These abstractions stand for alternative ways to intercept the perceived signal. Each agent applies an inner discretization strategy. The discretization can evolve in real-time in order to adjust the perimeter of the (states) classes the agent is looking for. Several mechanisms can increase the ability of the agents to fine-tune their discretization. The multi-agent processing leads to concurrent sets of partitions available for the system as possible representations. The action selection relies on a reinforcement process that make use of all the concurrent representations. The model, described in Algorithm 1, is expected to dynamically combine the previously learned perception-action states to build a more accurate resulting representation of the environment.

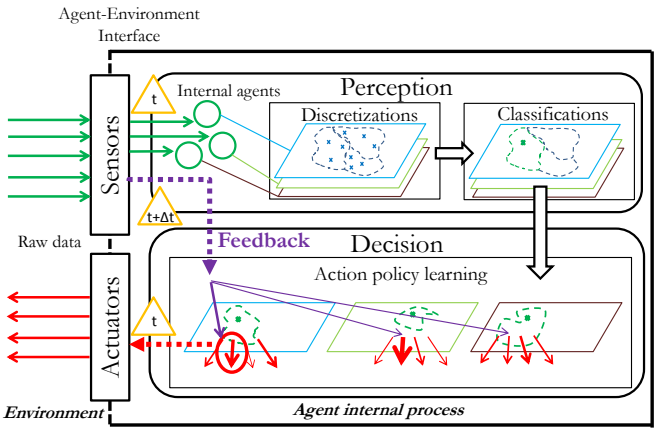


Fig. 2: General model overview.

Algorithm 1 Main execution loop of the system

Instantiate the set A of possible actions
Instantiate the set D of discretizer agents
Instantiate the set S of concurrent selected states
loop
  if S is not empty **then** *# bypass for first interaction*
    $F = feedback(P)$
    for all $S_i \in S$ **do**
      # get the link L between S_i and A^*
      $L = getPerceptionActionLink(S_i, A^*)$
      $incrementReward(L, F)$
    end for
    $reinforce(D, A^*, F)$
  end if
  $P = getPerception()$ *# action selection process start*
  $S.removeAll()$
  for all $D_i \in D$ **do**
    # agents perceive different variables of the perception
    $P_i = discretizerPerception(P)$
    $D_i.discretize(P_i)$
    $S_i = D_i.classify(P_i)$
    $S.add(S_i)$ *# add the selected state hypothesis S_i*
  end for
  $A^* = explorationExploitationActionSelection(S)$
  $execute(A^*)$ *# execute the action A^* in the environment*
end loop *# wait for the next perception (and feedback)*

B. Building a high-level perception

The system relies on sub-representations from internal discretizers agents in order to build a high-level representation. Each discretizer has its own classification process carried out in its perception space, that may vary from an agent to another. Our proposal enables an evaluation of the representations based on different input streams. Then, distinct discretizers could get inputs from the same sensors or uncorrelated inputs. We propose a generic description of the model which allows to choose the classification method during the model implementation. Both online or off-line classification methods are suitable provided that they accept the given inputs and that they generate a state-space representation. In order to enhance the search space exploration capabilities of the agents, each discretizer works on a sub-partition of the perception

stream. The variables set perceived by each agent can be expertly defined or randomly initialized, depending on the tackled problem. The perception process of the system can be described in two consecutive steps (steps 1 and 2), as represented in Figure 3:

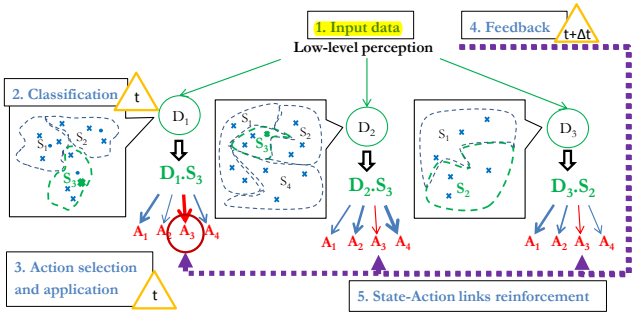


Fig. 3: Discretization, classification and action selection processes of the model.

Each time the system gets a perception, all the discretizer agents D update in parallel their space of potential states (hypotheses), thanks to a considered clustering method (K-means, SOM, etc.). For instance, the discretization of a hypothetical agent D_i could be a set composed of the following state hypothesis: $\{S_1, S_2, ..., S_j\}$. At each perception step of the system, each agent D perceives the inputs and classifies its perception as a discrete state hypothesis. Relying on the proposed hypothesis, the system tries to select the best action to perform under the context defined by different agents.

C. Controlling the environment

Discretizers memorize the probability of selecting an action when they classify their perception as a given state hypothesis. These probabilities are supported by state-action links and aim at providing the system with a memory of the result, evaluated through the feedback function, of past actions. After the perception stage, all the discretizers select one of their own state hypotheses in response to the propagated low-level perception. The system has to select the best possible action under the given hypothesis (one per agent). We are looking for a trade-off between exploration (considering the most valuable action) and exploration (attempting an action never tried so far), as represented by steps 3-5 in Figure 3.

The system faces a traditional exploration-exploitation dilemma [18] since the action space is supposed to be finite and there is no apparent correlation between distinct resulting state hypotheses. Such a problem is also known as a multi-armed bandit problem, which relates to the optimal strategy to select and test slot machines in a casino. Several strategies can be found in the literature and then included in the proposed model. Most of the existing approaches focus on minimizing a parameter called "regret" which can be seen as the difference between the current reward versus the optimal (theoretically unknown) reward. Our proposed model does not require such a strict optimal policy since under-exploring the state-action space could lead to bad global performance. However, we can assume that this consideration also strongly depends on the selected problem and has to be investigated carefully during the model implementation.

The feedback provided by the system is part of the low-level perception and has to be defined during the formalization of the problem (*i.e.* for implementation). This function allows the system to estimate the result of the previous action in the environment. The acquired value is used as a reward of the exploration-exploitation algorithm and contributes to the reinforcement of the constructed representations. The result of the selection is a state-action link and each state is directly linked to a given discretizer. Hence, reinforcing a given link implies reinforcing the corresponding agent. Thus, this reinforcement process can be seen as the learning process of the system which helps identifying the bests state hypotheses within the set of sub-spaces discretizations provided by the population of discretizer agents.

III. Application to Road Traffic Control

The generic model proposed in this paper has been implemented for an innovative application: cooperative traffic control. The simulation framework used for the experimentation has been presented in [19]. In this paper, we detail the implementation of the model and provide the description of the discretizers agents characteristics and the used feedback/actions. Then, results highlight the behavior of the model and confirm the expected benefits of using concurrent representations for controlling a flow of partially equipped vehicles.

A. Case study: Cooperative Traffic control

We select the case study of cooperative traffic control to implement our model and investigate the expected benefits of the proposed approach. A mixed flow of connected and non-connected vehicles evolves on a road network. Connected vehicles can share information with each other (V2V communication) and with the infrastructure (I2V /V2I). The infrastructure is composed of Road Side Units (RSU), in charge of controlling their dedicated section. A RSU can perceive information in order to estimate the current traffic state through its sensors (for instance, loop sensors which estimate the flow and speeds on a single lane). The objective of the RSU is to propagate recommendations to reachable connected vehicles. Such control strategies intend to reduce traffic jams.

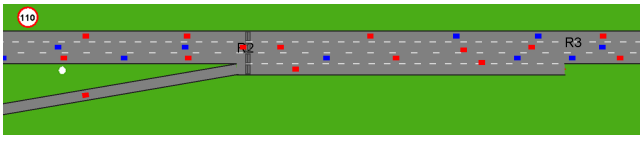


Fig. 4: Screenshot of the scenario within the simulator.

As depicted in Figure 4, all the experimentations will be conducted on a single scenario occurring on a 3 km straight section of a three-lane highway with an on-ramp near the end. The RSU (represented by the white disc) perceives flows and speeds through the linked loop sensors (gray rectangles on

the road) and connected vehicles speeds in its communication range (150 meters radius). The RSU implements the model and intends to give best recommendations to connected vehicles in order to avoid congestion, or at least, reduce it. Recommendations are propagated to reachable connected vehicles through I2V communication. The on-ramp triggers a sudden speed change from the right-most lane under high density conditions, and may form congestion on the three lanes. We investigate the effects of the RSU control strategy on traffic flow, assuming a fixed percentage of 30% of connected *vs.* non-connected vehicles, homogeneously mixed on the main section.

1) Simulation: The simulated scenario is modeled thanks to a multi-agent extension [20] of the MovSim simulator [21] (the Multi-model Open-source Vehicular-traffic SIMulator). This simulation platform implements several microscopic models (also known as car-following models) that describe the longitudinal behavior of vehicles. A graph-based representation of the road network allows to model multi-lane sections and includes speed limits and connections. Our extension focuses on the development of vehicles communication capabilities and provides a generic implementation of the infrastructure (RSU). A message-based protocol allows all the entities, modeled as autonomous agents, to share information and adapt their behavior.

2) Scenario: In order to generate several distinct states for the RSU, we use real data from detectors to design the input flows (main section and on-ramp) over time. The dataset is composed of one month (June, 2001) of loop sensor data for the 8:00 to 9:00 AM period in the French A6 highway. We select two detectors where the road network was matching the scenario configuration (Figure 4). Then, we were able to produce 30 simulations runs of 60 minutes each. The diversity of the simulated traffic situations allows to investigate the behavior of the model in sightly different contexts (free flow, critical regime, and congestion).

B. Behaviors

The integrated multi-agent framework enables a definition of complex behaviors for both the infrastructure and the vehicles. We propose to implement our model as the decision process of the RSU. In order to reproduce traffic flow dynamics, car-following rules model the longitudinal motion of the vehicles. This base behavior is overridden by recommendations received by connected vehicles.

a) Road Side Unit: The result of the control strategy of the RSU is the selection at each decision time step (set to 120 seconds) of an action among the following set:

- A_1: No action.
- A_2 – A_3: Lane changes (right to left – left to right).
- A_4 – A_5: Inter-distance changes (1.8 s – 1.2 s).
- A_6 – A_7: Speed limit (110 km/h – 50 km/h).

All the actions lead to the propagation of a recommendation message to connected vehicles containing the corresponding information. Such a message has an embedded relevance area (set to the whole section from the start until the on-ramp) and an expiration time (also set to 120 seconds). The recommendation held by a message is applied by a vehicle while it is in the relevance area and while the message has not expired.

b) Vehicles: The longitudinal behavior of connected and non-connected vehicles is simulated by a car-following model. We choose the Intelligent Driver Model (IDM) which is known as being collision-free and well reproducing instabilities of traffic flow [21]. Lane-change decision is modeled using an opportunistic strategy called MOBIL [22] (Minimizing Overall Braking Induced by Lane Changes). Only connected vehicles are equipped with a wireless communication device allowing them to perceive messages within a range of 150 meters (radius), matching the current technologies capabilities. We do not assume partially received message or loss of information due to the communication layer in this work, since the dedicated protocol for vehicle communication is expected to almost reach this assumption. Recommendation messages received by connected vehicles are rebroadcasted to other reachable connected vehicles until the beginning of the 3 km section.

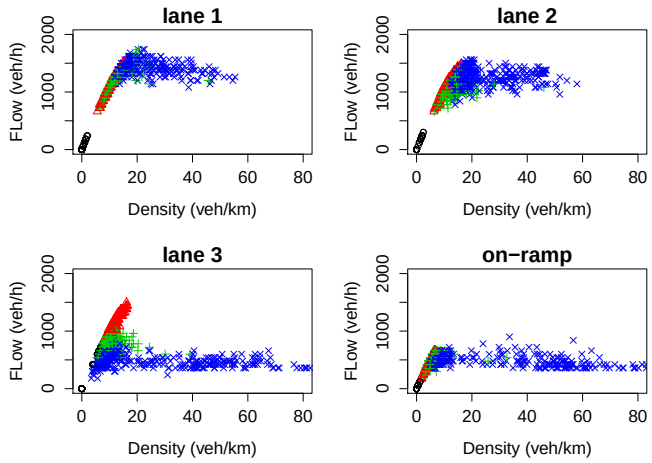
Connected vehicles are displaying received recommendation messages on an inboard interface allowing the driver to adapt its behavior to current recommendation. Thus, we propose an implementation of the interpretation of the messages by the vehicles. For actions A_2 and A_3, the lane-change recommendation overrides the opportunistic strategy as long as the message remains relevant. It means that the vehicle will try to reach the most-left lane (respectively most-right lane) by intending lane-change since reaching the targeted lane. The lane-change model gap estimation is used to validate if an intention of lane-change is safe enough (in terms of the parameters set), to be effectively executed. Basically, it means that a higher proportion of connected vehicles will actually change lane when a lane-change recommendation is propagated when the traffic density is lower (and vice versa). For actions A_4 to A_7, the only modification is to temporarily modify the corresponding vehicle longitudinal parameter (respectively "desired time headway" and "desired speed") to match the given recommendation. All these adaptations are maintained during the relevance duration of the received recommendation message and are automatically canceled if the message expires or if the vehicle leaves the relevance area.

C. Model implementation

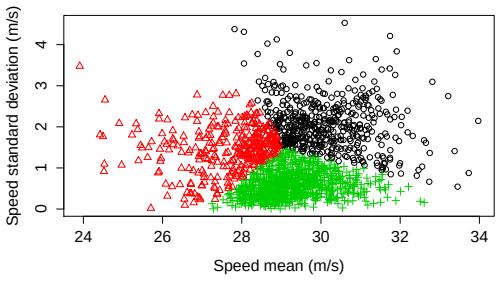
Under the case study of cooperative traffic control, we propose to implement the model presented in this paper as the decision process of a Road Side Unit. The RSU is linked to a loop sensor which computes the mean flow, density and speeds on the on-ramp and on the three lanes of the section. In addition, connected vehicles driving in the communication range of the RSU share their respective speed. This data is used as the low-level perception of the system. Then, multiple discretizers agents are in charge of the construction of a representation lying on state hypotheses. In this work, the system perception and decision process is discretized in 120 seconds time periods. The RSU perceives data from its sensors from the last 120 seconds, and immediately choose

and execute an action (*i.e.* send a recommendation). After 120 additional seconds, the recommendation expires and the RSU receives a feedback to reinforce the representation, and at the same time, gets the new perception of the environment.

1) Discretizers instanciation: In order to show the benefits of the model, we choose to implement the model using 2 distinct discretizers. The purpose of each discretizer is to generate an individual representation from its perceived data. We decide to link each discretizer, D_1 and D_2, to a specific sensor. The main difference between the two agents is their perception. D_1 is linked to the loop-detector and perceives the mean flow and the mean density for each of the three lanes plus the on-ramp, aggregated each 120 seconds. D_2 exploits data from the RSU's communication device and builds a model (mean and standard deviation) of the estimated speed distribution from the messages shared by the connected vehicles passing by during the last RSU decision time step. The discretizations results from a K-means clustering (stabilized with 10000 iterations with random initializations) with respectively 4 and 3 classes (for D_1 and D_2). The results of the clustering by both agents on the data from the first 15 simulation runs are depicted in Figure 5.



(a)



(b)

Fig. 5: Alternative state partitions generated by (a) D_1 [8 dimensions] and (b) D_2 [2 dimensions] on the training set.

Learning of the perception-action states of each agent was performed on the same 15 simulations, exploiting the feedback provided by the system.

2) Feedback: Rewards gathered by the discretizer agents in state-actions links are computed for a selected input from the environment. In the proposed scenario, we select the minimal observed mean speed (on the four detectors) as the feedback provided by the system, since it matches the objective: globally improve the flow. This external feedback is used as a reward in the exploration-exploitation algorithm for reinforcing both the control strategy and the proposed representations. Among the multi-armed bandit algorithms proposed in the literature, we choose to implement Upper Confidence Bound [18] (UCB). This algorithm initialization is to try each machine once. Then, the algorithm systematically chooses the machine j that maximizes $\ddot{x}_j + \sqrt{\frac{2 \ln n}{n_j}}$ where $\ddot{x}_j$ is the estimated average reward obtained for machine j, n_j is the number of time machine j has been played so far, and n is the overall number of plays done so far. The rewards update concerns all tuples (links between perception-action, denoted as L in Algorithm 1) containing the selected action, not only the tuple that is selected by the algorithm. This is a way to propagate the benefit of an action selection on all the tuples that contributed to the action selection. Then, the frequency of selection which is increased only for the tuple that has been selected by the algorithm. This reward strategy speeds up the exploration process, thanks to considering the update of as many links as discretizer agents, at each iteration step.

D. Experimentations

The experimentations are conducted following two main steps. First, a set of input data (D agents' perceptions) is needed to run the clustering algorithms and obtain a stabilized classification (without applying action propagation). Then, a training set of simulation is used to make the system complete its representation through the whole process of exploration and action testing (first 15 simulations). Secondly, the system is ready to be confronted to alternative scenarios, expecting that the built representation helps the system to target a beneficial control strategy on the flow. This phase has been conducted on the second half of the simulations set. The baseline of each simulation was the observation of several indicators without using the infrastructure (meaning that neither recommendation nor communication is used). Then we test three implementations of the model: the single agent case (D_1 and D_2 separately), and a dynamically built combination of the representations from the two agents D_{1-2}.

1) Indicators: In order to assess the effect of the different strategies, we have selected 3 indicators: the Total Time Spent (TTS), the section mean speed and the percentage of congestion. The TTS is basically the sum of the travel time of all the vehicles on the section; lower values means better efficiency. The mean speed allows to observe variation during time, and hence the flow homogeneity. The percentage of congestion is the spatial proportion of the section where vehicles speed are below a threshold (set to 30 km/h). We expect that the model implementation will make use of the individual representation of the agents to improve the traffic flow, observed through these indicators.

2) Results: Among the 15 available simulations, we selected 3 runs that illustrate different traffic situations. Figure 6 depicts a plot of the indicators for the 3 runs.

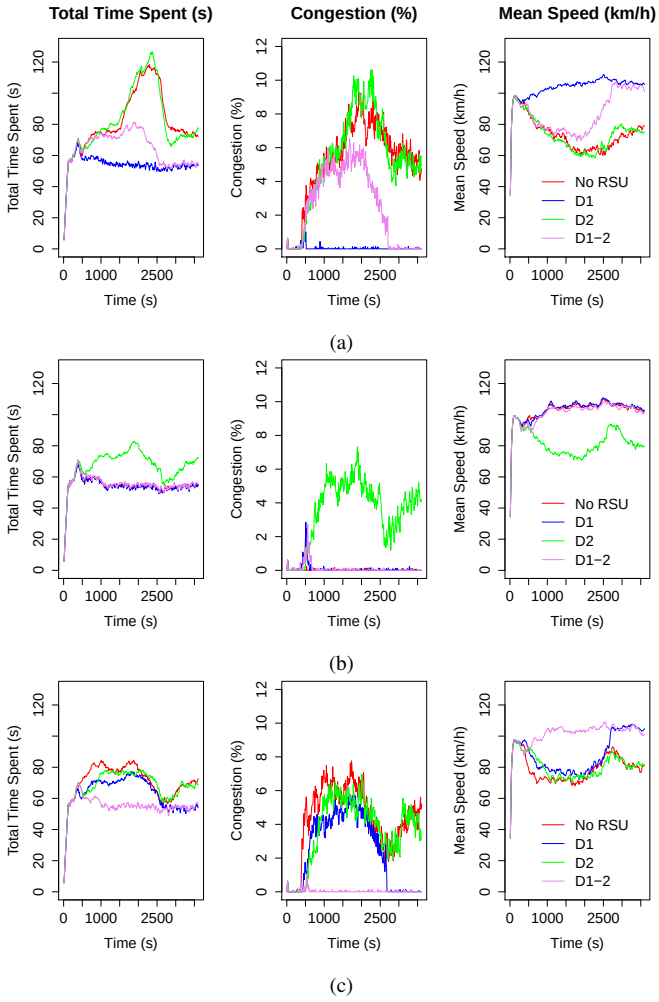


Fig. 6: Results of 3 simulations: (a) Monday 18, (b) Thursday 21 and (c) Tuesday 28.

The indicators allows to evaluate the benefits of each implementation of the model, in comparison to the baseline (red curves). In the simulation (a) and (c), without control, the instabilities introduced by the lane-changes from the on-ramp are propagated to the the flow. The three simulations show different behaviors of the model. In simulation (a), the control strategy of agent D_1 avoid the apparition of the congestion whereas the one from agent D_2 globally perform worst than the baseline. These results are clearly visible on the TTS indicator. Decreasing the Total Time Spent means that the overall vehicles travel time is shorter. The dynamic combination of D_1 and D_2 produced a balanced control strategy which results in a mitigate improvement. Simulation (b) highlights a specific case where one of the discretizer representation used in the model fails (D_2 strategy), since no congestion is observed in the baseline. Nevertheless, the model has been able to converge to the best individual representation (D_2) in this context, which does not improve the traffic state observed in the baseline. In Figure 6 (c), the modeled traffic state is near to the flow equilibrium, since the congestion is limited and the mean speed does not drop too much (comparatively to simulation (a)). In this context, both strategies learned by the discretizers have a positive impact on the flow, and the ability of the model to enhance its representation is visible. The system has used the rewards gathered using the feedback to build a merge representation that can avoid the propagation of congestion on the section.

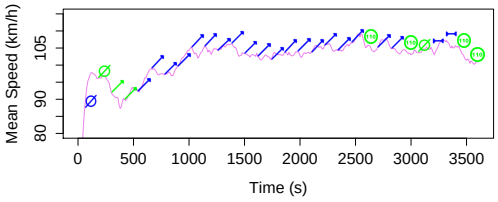


Fig. 7: Actions taken from the internal agents by the model for simulation (c).

Figure 7 displays a simplified representation of the contribution of each discretizer agent (using the same colors as in Figure 6) in the system control strategy. As expected, both of the perception-action states of the two discretizer agents are used by the model and the most used action (A_2) sounds logical since it helps the vehicles on the on-ramp to merge in the main section. Hence, the results confirm the ability of the model to make use of concurrent representations and merge them into an improved representation.

3) Discussion: The conceptual definition of the proposed model implies that the system designer needs to be careful during the implementation. Indeed, as in most of the reinforcement learning problems, choosing a relevant feedback can be challenging. Moreover, since the system relies on concurrent discretizations, the input used for the perception of the agent is very important. This is related to a sensor fusion issue: some phenomena can be more easily enhanced using the appropriated dimensions. It means that the system has to perceive relevant inputs in order to be able to adapt the control to the modifications of the environment. As seen in the experimentations, the system designer has to provide a first set of discretizer agents that can provide good state identifications, before intending to combine them. This issue could be tackled by providing the model with mechanisms for autonomously creating and making evolving discretizer agents, taking advantage of previously learned perception-action states.

As the inputs can come from several dimensions (or variables), one can try to provide the system with a full perception (using all the input data). When implementing the model, exploring the whole set of combinations of input variables for one single discretization would be very costly in terms of exploration time. This is why we propose to use several agents, that consider sub-sets of the low-level perception. We

can expect from the results obtained in the experimentations a convergence of the system (in term of perception-action link profile) toward combinations of input variables that are relevant according to the system use (*i.e.* regarding the resulting control strategy). This will be particularly useful for problems where combinations of inputs are not part of prior knowledge. Moreover, by dynamically combining parts of the perception-action states provided by discretizer agents, we can imagine that the system will explore relevant combinations linked to the actions performed, and hence resulting in a sparse association of combinations in different subspaces of perception. All these improvements will be part of model evolutions. Indeed, a further step will deal with associations of perception-action states linked to the control of the system.

IV. Conclusion and further work

We have proposed a generic model able to construct a representation from its interaction with the environment. As a constructivist effort to build iteratively the decision making process of control systems, the proposed model makes use of a population of discretizer agents to bootstrap the construction of the representation. The resulting representation is composed of perception-action states which are part of a reinforcement learning process using a feedback from the environment to explore the search space. After presenting the model from a theoretical standpoint, we propose an application to the case study of road traffic control. The challenge is to make use of future connected vehicles and infrastructure in order to build an autonomous dynamic decision-making system. This innovative application illustrates the benefits of the approach. The obtained results from simulation show that the combination of individual discretizations provide with the system an improved control strategy, taking the best of the individual strategies.

Further work will be mainly focused on the evolution of the model. We have assumed an iterative construction of the conceptual model, as this ensures to test and validate the mechanisms at each stage through a robust implementation environment. Next step will be to design a new kind of agent in charge of exploiting the proposed states hypotheses to dynamically create states associations. Basically, two types of associations are possible between generated states: aggregations or specializations. The main questions that arise are about the trade-off between exploration time and the possibility of combining previously learned state-action links. In other words: how could we combine efficiently existing states to catalyze the new state-action links versus how could we ensure an optimal exploration to the system? We plan to investigate collective artificial intelligence methods to tackle the problem of the combination of perception-action states. These methods will also be useful to provide the system with the ability to dynamically create new discretizations from relevant inputs combinations.

Acknowledgment

The work presented in this paper was supported by a grant from la Région Rhône-Alpes.

References

[1] R. A. Brooks, "Intelligence without representation," *Artificial intelligence*, vol. 47, no. 1, pp. 139–159, 1991.

[2] J. Piaget, "The construction of reality in the child," *Journal of Consulting Psychology*, vol. 19, no. 1, p. 77, 1955.

[3] J. Zlatev and C. Balkenius, "Introduction: Why "epigenetic robotics"?" in *Proceedings of the First Conference on Epigenetic Robotics. In: Balkenius, C., Zlatev, J., Kozima, H., Dautenhahn, K., Breazeal, C.(eds.): Lund University Cognitive Science Series*, no. 85, 2001.

[4] F. Guerin, "Learning like a baby: a survey of artificial intelligence approaches," *The Knowledge Engineering Review*, vol. 26, no. 02, pp. 209–236, 2008.

[5] J. Mugan and B. Kuipers, "Autonomous representation learning in a developing agent," in *Computational and Robotic Models of the Hierarchical Organization of Behavior*. Springer, 2013, pp. 63–80.

[6] B. J. Kuipers, P. Beeson, J. Modayil, and J. Provost, "Bootstrap learning of foundational representations," *Connection Science*, vol. 18, no. 2, pp. 145–158, 2006.

[7] J. Provost, B. J. Kuipers, and R. Miikkulainen, "Self-organizing distinctive state abstraction using options," in *Proc. of the 7th International Conference on Epigenetic Robotics*, 2007.

[8] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. Touretzky, and T. Leen, Eds. MIT Press, 1995, pp. 625–632.

[9] T. Kohonen, *Self-Organizing Maps*, ser. Springer Series in Information Sciences, H. N. Y. Springer, Berlin, Ed., 1995, vol. 30.

[10] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. pp. 100–108, 1979.

[11] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[12] A. Kesting, M. Treiber, M. Schönhof, and D. Helbing, "Adaptive cruise control design for active congestion avoidance," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 668–683, 2008.

[13] R. C. Carlson, I. Papamichail, and M. Papageorgiou, "Local feedback-based mainstream traffic flow control on motorways using variable speed limits," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1261–1276, 2011.

[14] A. L. Bazzan and F. Klügl, "A review on agent-based technology for traffic and transportation," *The Knowledge Engineering Review*, vol. 29, no. 03, pp. 375–403, 2014.

[15] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. IEEE Computer Society, 2004, pp. 530–537.

[16] M. Vasirani and S. Ossowski, "A market-inspired approach to reservation-based urban road traffic management," in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 617–624.

[17] C. Canudas de Wit, "Best-effort highway traffic congestion control via variable speed limits," in *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*. IEEE, 2011, pp. 5959–5964.

[18] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[19] M. Guériau, R. Billot, N.-E. El Faouzi, S. Hassas, and F. Armetta, "Multi-agent dynamic coupling for cooperative vehicles modeling," in *The Twenty-Ninth Conference on Artificial Intelligence AAAI'2015 - (DEMO Track)*, 2015, pp. 4276–4277.

[20] M. Guériau, R. Billot, N.-E. El Faouzi, J. Monteil, F. Armetta, and S. Hassas, "How to assess the benefits of connected vehicles? a simulation framework for the design of cooperative traffic management strategies," *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 266 – 279, 2016.

[21] M. Treiber and A. Kesting, *Traffic flow dynamics: data, models and simulation*. Springer, 2013.

[22] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1999, no. 1, pp. 86–94, 2007.