SIMULATION STUDY OF LEARNING AUTOMATA GAMES IN AUTOMATED HIGHWAY SYSTEMS

Cem Ünsal
Robotics Institute, Carnegie Mellon University
Pittsburgh, PA, 15213-3890 USA

Pushkin Kachroo, and John S. Bay
The Bradley Dept. of Electrical Engineering, Virginia Tech
Blacksburg, VA, 24061-0111 USA

ABSTRACT

We propose an artificial intelligence technique called stochastic learning automata *to design an intelligent vehicle path controller. Using the information obtained by on-board sensors and local communication modules, two automata are capable of learning the best possible actions to avoid collisions. Although the learning approach taken is capable of providing a safe decision, optimization of the overall traffic flow is required. This can be achieved by studying the interaction of the vehicles. The design of the adaptive vehicle path planner based on local information is extended with additional decision structures by analyzing the situations of conflicting desired vehicle paths. The analysis of the situations and the design of these structures are made possible by treatment of the interacting reward-penalty mechanisms in individual vehicles as automata games.*

1. INTRODUCTION

One of today's most serious social, economical, and environmental problems is the traffic congestion. To increase highway safety while reducing congestion, US Department of Transportation has taken an approach called the Intelligent Transportation Systems (ITS). A major element of ITS development effort is the Automated Highway Systems (AHS). Vehicle control is probably the most important part of the advanced AHS applications, because technological requirements of such a system are well beyond human capabilities. A large group of investigators is working on vehicle control issues [2]. However, being able to control vehicle dynamics does not necessarily mean that we have an AHS. In an environment with many fast-moving vehicles, making the right decision to avoid collisions and optimize the vehicle path is difficult. Initial research on automated vehicle control indicates that a planning system that can guarantee optimal operation with a sound theoretical background has not yet been developed, and it may be vital to AHS implementation [2].

We visualize two learning automata employing a reinforcement learning algorithm as the heart of our path planner. Using local sensor and limited communications data, the automata learn the optimal actions to be taken for a given situation. Given enough time and correct learning parameters, the automata indicate the best actions to take, and send these actions to the lower control layer. The initial decision system uses mainly local information, and consequently, the actions learned by the intelligent controller are not globally optimal; the vehicles can survive, but may not be able to reach some of their goals. To overcome this problem, we treat pairs of automata as interconnected automata structures and visualize the interaction between vehicles as sequences of games played between automata. By evaluating these games, it is possible to design new decision rules, and to analyze the interactions between vehicles.

2. A LEARNING METHOD FOR NAVIGATION

Recent research on intelligent vehicle includes adaptive intelligent vehicle modules designed to answer the need for real-time maneuver selection for tactical driving [5]. Another approach to intelligent control for autonomous navigation uses a decision-theoretic approach with probabilistic networks where the problem is modeled as partially observable Markov process, and the optimal action is a function of the current belief state [1]. Similarly, a rule-based navigation system that uses worst-case decision-making is defined in [4]. Our approach differs from the above-mentioned works in the use of learning paradigm. Instead of learning the parameters affecting the firing of actions on repeated runs, the automata learn which action to fire based on the local sensor information. In other words, the higher level learning is not in the design phase, but in the *run* phase. There are no "prescribed conditions" for actions. The idea of defining a "fixed" structure to be utilized to find the optimal action has its own appeal, since the performance of the system is deterministic in the sense that the best action for a specific situation is known. However, drivers do not follow rules deterministically. In this sense, the learning automata approach is able to capture the dynamics of driver behavior.

A crucial advantage of learning compared to other learning approaches is that it requires no information about the environment except for the reinforcement signal. The learning paradigm of the stochastic automaton is based on repeated actions and the resulting

environment responses. One action is selected based on the action selection mechanism, the response (favorable or unfavorable) from the environment is observed, then the action selection mechanism is updated based on the response, and the procedure is repeated. The algorithm that guarantees the desired learning process is called a *reinforcement scheme*. Learning automata and reinforcement schemes are exclusively investigated during the last few decades [3].

2.1 Learning Automata as Intelligent Controller

For our model, we assume that an intelligent vehicle is capable of two sets actions. Lateral actions are *shift-to-left* (SL), *shift-to-right* (SR) and *stay-in-lane* (SiL). Longitudinal actions are *accelerate* (ACC), *decelerate* (DEC), *and keep-same-speed* (SM). The actions SiL and SM are "idle actions," and can be treated as a single action. An autonomous vehicle must be able to 'sense' the environment around itself. Furthermore, it must have the knowledge of its own displacement. Therefore, we assume that there must be a minimum of four basic sensors on board the vehicle: the *headway sensor*, two *side sensors*, and a *speed sensor*. The headway sensor is a distance-measuring device that returns the headway distance to the object in front of the vehicle. Side sensors are able to detect the presence of a vehicle traveling in the immediate adjacent lane. The speed sensor is simply an encoder returning the current wheel speed of the vehicle.

Each sensor is connected to its associated decision module that specifies an output signal in response to environmental data. In addition to these sensors, there may be two additional modules as shown in Fig. 1: a *lane* sensor, and a *pinch* module described in the next section. Sensor and communication modules evaluate the sensor signals in the light of current actions, and send a response to the automata. The feedback is a combination of the outputs of all sensor modules.

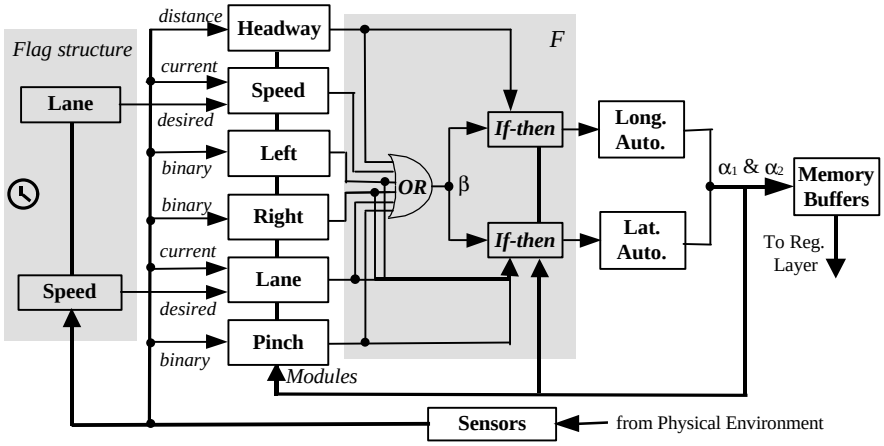

Figure 1. Learning automata in a multi-teacher environment connected to the physical layers.

It is important to differentiate between the "automaton environment" and the "physical environment." The output α of an automaton is a signal that defines the current choice of action. It is the lower control layer's (described as *regulation* layer in [7]) responsibility to interpret this signal. When an action is carried out, it affects the physical environment. The sensors in turn sense the changes in this environment, and the feedback loop is closed with the sensor modules and the response signal β. The regulation layer is not expected to carry out the chosen action immediately. Only an action that is recommended *m* times consecutively by an automaton is carried out. When this buffer is filled with the same action, that action is fired. After an action execution, memory buffers are filled with idle actions (SiL or SM).

2.2 Sensor modules

The four basic sensor modules listed above are simple decision blocks that calculate the response associated with the corresponding sensor, based on the last chosen action. For example, a penalty response (indicated by '1') from the left side sensor is received only when the action is SL *and* there is a vehicle in the left sensor's range (or the vehicle is already traveling in the leftmost lane). All other situation-action combinations result in a reward response ('0') from the left sensor module. The front sensor parameters describe the distance under which the presence of a vehicle is not desired. If the sensor "sees" a vehicle at a relatively close distance, a penalty response is sent to the automaton for actions SiL, ACC, and SM. All other actions (shifting lanes and decelerating) may serve to avoid a collision, and therefore, are encouraged. Again, the evaluation of the front sensor response based on the headway distance (and its rate of change) can be more complicated than the one described here. The speed module's task is to compare the actual speed to the desired speed. When the actual vehicle speed differs from the desired speed by more than a predefined amount, the action that will decrease the speed deviation receives a reward; others are penalized.

The additional lane detection module is used to make optimal path decisions as we describe later. A physical implementation of this module could be a vision system. For our purposes, we will assume that an automated vehicle can sense its present lane, and that it has some idea about where it should be. Based on these two values, the action that leads to the necessary lane shift is encouraged by this teacher module.

It is imperative for an automated vehicle to make sure that the adjacent lane is not "claimed" by another vehicle before changing to that lane. The "pinch condition" occurs when two vehicles one lane apart shift to the same spot in the lane between them. In our simulations, we use the memory vector to check for other vehicles' intentions to shift lanes. If an 'intention' signal is received from a neighboring vehicle, the pinch module sends a penalty response for the lateral action that may cause a problem. In a sense, the pinch module in an automated vehicle is driven by the memory vector of neighboring vehicles.

The flag structures shown in Fig. 1 are defined in order to obtain a more optimal trajectory by temporarily altering the behavior of the vehicle. The *lane flag* enables the automated vehicle to take action if it cannot reach its desired lane in a predefined time interval. If the vehicle cannot change to its desired lane in time, then the lane flag is set. The effect of this flag is to temporarily change the value of the desired speed. As a result, the vehicle slows down (or speeds up) in the hope of an opening to change lanes. Once the vehicle reaches its desired lane, the flag is reset.

Another flag to change temporarily the desired lane value is the *speed flag*. It keeps track of the elapsed time after the current speed deviates from its desired value for the first time. If the vehicle is unable to adjust its speed in the predefined time interval, then the speed flag is set. This flag forces the lane detection module to send a penalty response to the lateral action SiL, forcing the vehicle to change lanes if there is an opening. Detailed descriptions of the sensor modules and the flag structures as well as the complex reward-penalty mechanisms can be found in [6].

2.3 Learning Mechanism

Now that we have defined the sensor module outputs, the problem is to employ these signals for reinforcement learning. Sensor modules are separate teachers with possibly conflicting responses. The outputs of the six sensor modules described in the previous section are combined into a single response. As shown in Fig. 1, the function F that maps multiple teacher responses into a single feedback signal for each automaton, consists of an OR gate and two additional *if-then* condition blocks. Table 1 shows the possible responses from the teacher modules for each action. Since a penalty response ('1') will inhibit a reward response ('0') by using an OR gate, the mapping is almost complete except for one problem with longitudinal action DEC. If the headway module returns a reward for this action, this must inhibit a penalty from the speed sensor to guarantee safe operation. Furthermore, a penalty response to action SiL is inhibited by the longitudinal action DEC, for a smoother vehicle path.

Once we have the combined environment response for both automata, the control loop can be closed. As indicated before, a stochastic automaton learns from previous action and responses. Each action a_i of the automaton is assigned a probability p_i; the sum of all action probabilities is of course equal to 1. After an action a_i is executed, the response b is observed. The action probabilities are then adjusted according to this response. If the response is favorable, the probability p_i is increased; otherwise, it is decreased. The reinforcement scheme is a mapping of the action probability vector, automaton action, and the environment response at time step n to the next action probability vector at time step n+1. A variety of linear, nonlinear and hybrid schemes exists for stochastic automata [3].

Table 1. Action-sensor module response matrix.

	Hdway	Left	Right	Speed	Lane	Pinch
SiL	0/1	0	0	0	0/1	0/1
SL	0	0/1	0	0	0/1	0/1
SR	0	0	0/1	0	0/1	0/1
SM	0/1	0	0	0/1	0	0
ACC	0/1	0	0	0/1	0	0
DEC	0/0*	0	0	0/1	0	0

3. LEARNING AUTOMATA GAMES

The decision system in the previous section mainly uses local information, and as a result, the actions not globally optimal. The vehicles can survive, but may not be able to reach some of their navigational goals. To overcome this problem, we treat the interaction between vehicles as sequences of games played between pairs of automata. Every game corresponds to a "state" of the physical environment as described below.

The vehicle controller includes two automata, one each for lateral and longitudinal actions. There is a "direct interaction" between two automata in a vehicle due to the description of the teacher modules, and combination of the multiple teacher responses (Fig. 2).

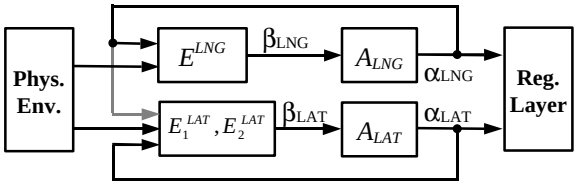

Figure 2. The longitudinal automaton determines the lateral automaton's environment (adapted from [3]).

Both automata update their action probabilities based on the responses of the environment. Furthermore, the value of the current longitudinal action changes the environment response to the lateral automaton. The idea of interacting automata was first introduced in [8]. The resulting configurations can be viewed as games of automata with particular payoff structures. We know that the lateral automaton A_{LAT} can operate in both lateral environments. In some situations, the choice of longitudinal action a_{LNG} affects the response of the lateral

environment. All other environment changes are due to the changes in the physical environment, and we visualize these changes as state transitions. Longitudinal automaton A_{LNG} is also capable of converging to its best action [3]. The lateral automaton A_{LAT} in turn would converge to the best action in the environment determined by A_{LNG}.

Assume that the probabilities of receiving a penalty from the environment for all actions are known. For example, for an automated vehicle that finds itself in the rightmost lane of a two-lane highway after merging from an entry, the lateral action SR will receive penalty until the vehicle shifts lane. Consider the situation in the first few seconds where the automata environment is stationary. (With relatively fast update rates, this assumption is always possible.) Provided that the vehicle is in its desired lane and speed range, the environment response for the actions depends on the output of the headway and the left sensor module. Assume further that the probabilities of sensing a vehicle in front and side sensor ranges can be calculated for this particular case. Then, by treating the probabilities of penalty as game payoffs for longitudinal and lateral automata, we can write the game matrix of penalty probability pairs:

$$\begin{array}{c} \\ ACC \\ DEC \\ SM \end{array} \begin{array}{c} \begin{array}{ccc} SL & SR & SiL \end{array} \\ \begin{bmatrix} \left(\frac{7}{30},\frac{4}{30}\right) & \left(\frac{7}{30},1\right) & \left(\frac{7}{30},\frac{5}{30}\right) \\ \left(0,\frac{4}{30}\right) & (0,1) & \left(0,\frac{1}{30}\right) \\ \left(\frac{4}{30},\frac{4}{30}\right) & \left(\frac{4}{30},1\right) & \left(\frac{4}{30},\frac{5}{30}\right) \end{bmatrix} \end{array}$$

First element of a payoff pair gives the probability of penalty for longitudinal action, while the second number is for the lateral action. Entries in the first and third rows correspond to the environment E_1^{LAT}; the second row is associated with E_2^{LAT}. The probability of penalty for lateral action SiL is less in the second environment where the longitudinal action is DEC. If the automata were not connected, an absolutely expedient reinforcement scheme would force the automata to converge to actions DEC and SL (lateral action SL will be optimal since if the penalty from the front sensor is not suppressed). Based on this payoff structure, the current solution pair (DEC, SiL) is Pareto optimal and is an equilibrium point for this game.

The interaction between automata is via the physical environment that is assumed to be stationary for the duration of a specific game. This results in a stationary automata environment, and the solution of such a disjoint game is an equilibrium point (and a Pareto optimal solution) due to the convergence characteristics of the reinforcement schemes.

While the two automata in each vehicle are guaranteed to reach the optimal solution for a stationary environment, interaction between vehicles creates another level of connection via the physical environment. The automata actions from other vehicles change the physical environment that in turn affects sensor module responses. This type of indirect interaction cannot be formulated using a game matrix. Furthermore, the fact that such a game matrix will be time varying when considering multiple interacting vehicles complicates the matter. Instead, we treat the automata environments resulting from the ever-changing physical conditions as a switching environment. Every state of the automata environment resulting from the changes in the physical world includes a different set of feedback responses for automata actions. These different states of the environment are assumed to be stationary if the automata converge to the optimal actions long before another change takes place. Once a decision is made and sent to the regulation layer, corresponding actions are fired and the changes in the physical environment force the automata environment to switch to another state.

The actions need not be fired for the environment to switch from one state to another. For example, the physical environment may change due to speed differences between vehicles while only the idle actions (SiL and SM) are fired consecutively. The moment that one vehicle clears another vehicle's sensor detection area, the state of the automata environment changes. The interaction between the actions and the physical environment, and the physical and automata environments, are fairly complicated. Here, we will introduce a representation scheme that will facilitate the analysis of changes in the physical world in relation to the automata environment. Illustrating vehicle interactions as automata games for every instance of the automata environment is not feasible, but it may be possible to define a similar matrix for all actions of autonomous vehicles. In a situation wherein autonomous vehicles interact via their sensors and communication devices, the physical presence of a vehicle affects the automata environment of another.

Consider the situation shown in Fig. 3a. where vehicle 1 and 2 are autonomous, but vehicle 3 is not. It is just an obstacle to the other vehicles. Vehicle 2 has no lane preference while vehicle 1 needs to shift to middle lane, but is unable to fire this action because vehicle 2 is in its side sensor range. The automata environment for this situation is given in Fig. 3b. (Actions SiL and SM are combined into a single action *IDLE*. If a lateral action other than SiL is chosen, the row/column for combined action IDLE refers to the longitudinal action SM, and vice versa.) Due to velocity differences, vehicle 2 drifts away from vehicle 1's sensor range, and the automata environment switches to a new state (triangles indicated by light gray are cleared; Fig. 3b). In the mean time, the idle actions are fired repeatedly. With the change, the number of possible actions for vehicles 1 and 2 increases,

and lateral action SL becomes the optimal solution for vehicle 1. Consequently, vehicle 1 changes lane, which in turn causes another automata environment change.

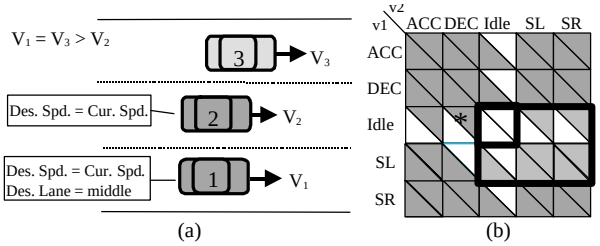

Figure 3. The physical and automata environments: The matrices give the conditions in a particular automata environment resulting from current conditions and vehicle parameters. If the combined response is a penalty, it is shown as a shaded triangle; rewards are shown as white triangles. Upper triangles are associated with Vehicle 2; optimal action pairs are indicated with black borders.

Using the same reasoning, we can establish which automata environment corresponds to each physical situation-vehicle condition pair. The convergence to the optimal solution is guaranteed for all such situations. It is then possible to predict how the vehicle will react to a specific physical situation. This will enable us to define *highway scenarios*, and find solutions for intelligent path planning.

4. ANALYSIS OF HIGHWAY SCENARIOS

Although vehicle controllers described above are able to avoid collisions, the resulting vehicle paths may not be the best solution for the problem of congestion. Some vehicle paths may also conflict and prevent the vehicles from reaching their desired goals. We visualize a possible situation with multiple vehicles as a sequence of environment states. For all the states of the physical environment –which includes the positions of the vehicles and current parameters defining their behavior– a corresponding automata environment can be defined. The automata environment is analyzed to predict possible physical environment changes. These changes are illustrated as state transitions. State diagrams formed using possible environment state transitions can be then used for analysis as well as design purposes.

Consider two vehicles sharing a 3-lane highway. All possible physical situations that arise while considering two vehicles in a three-lane highway are simplified to 12 states in Fig. 4. We assume that only three possibilities exist for relative longitudinal positions. The distinguishing factor between these positions is the sensor readings. Each row in a matrix corresponds to a lane; each dark square indicates the presence of a vehicle in a road segment covered by side sensors. Not all possibilities are considered; instead, only the situations that are of interest for a specific scenario will be represented. Similar situations are also combined into a single state and simplified if necessary. Two situations are said to be similar if the sensor module outputs and/or possible actions are the same for both. Note that for each state given in Fig. 4, there is a reciprocal state with switched vehicle positions, denoted by an asterisk.

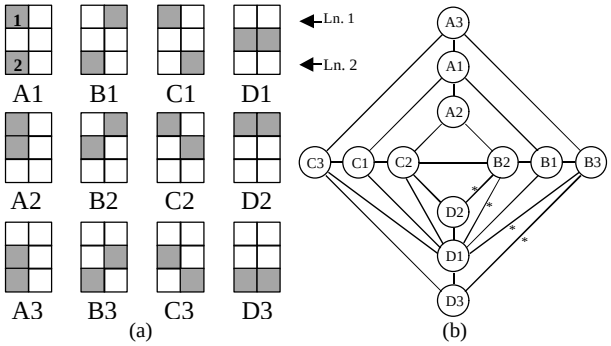

Figure 4. (a) Possible physical environment states for 2 vehicles in a 3-lane highway, and (b) state transition diagram for these states (self-transitions" are not shown; "*" indicates a transition to a reciprocal state).

To analyze the behavior of autonomous vehicles and the conflicts resulting from their interactions, we define highway scenarios that combines physical location, sensor outputs, and internal parameters of vehicles. Once we know the automata environment at the beginning of a scenario, we can predict the changes in the physical environment. Then, all possible changes are combined to form a state transition diagram showing the progression of the physical environment. The transitions between states are the direct results of the automata environment described by the matrices such as those given in Fig. 3.

Now, consider the situation A1 with two intelligent vehicles equipped with sensor modules. The velocities and lateral positions of the vehicles are the same. Suppose vehicle 1 needs to shift to lane 3, and vehicle 2 to lane 1. Since the vehicles are traveling at the same speed, there are no actions that would lead to a goal state using the basic sensor modules. Possible transitions are A1→A2 and A1→A3. For transitions to these states, one of the vehicles must fill its memory vector with a lane shifting action.

If the vehicles were to change speed, multiple transitions leading to goal states are possible. Suppose that vehicle 2 decelerates. Then automata environment shown changes with the deletion of the penalty response for action DEC (Fig. 4b). The physical environment will switch at some point, due to the longitudinal actions taken. Therefore, the transitions of Fig. 5 are possible.

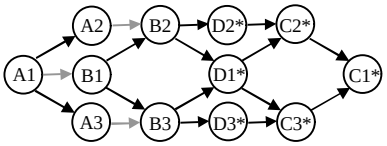

Figure 5. Possible transitions if vehicle 2 is able to decelerate.

All transitions in Fig. 5 except those indicated by gray color are automatic under the current circumstances. (There are other possibilities solving the deadlock situations depending on the permitted longitudinal

actions for vehicle 1 and 2.) To introduce the change to the automata environment, the lane flag module described in Section 2.2 is designed. Of course, the forced speed change must be different for left and right lane changes in order to break the symmetry.

Consider a similar situation with three automated vehicles on a 3-lane highway (Fig. 6a). Again, the solution lies in changing the relative speeds of the vehicles. The state transition leading to a solution is in Fig. 6b. A few other solutions are also possible if different speed adjustments are considered.

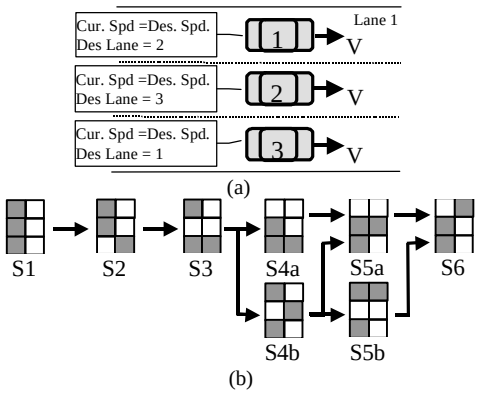

Figure 6. (a) Three vehicles with conflicting paths, and (b) a possible chain for this scenario.

All transitions except the first one are automatic given current vehicle parameters. For the first transition, on the other hand, the lane flag needs to be set in at least one vehicle. The problem and the solution for this case are similar to the 2-vehicle scenario. This is not a coincidence; it is due to the *superposition* of the two 2-vehicle situations. The term 'superposition' indicates that a 3-vehicle situation can be treated as three separate asynchronous 2-vehicle interactions (Fig. 7). The transitions that need to be forced by the lane flag are (and must be) between corresponding states in 3-vehicle and 2-vehicle transition diagrams. It is possible to define complex situations of multiple interacting vehicles as a group of many 2-vehicle situations. A complex scenario is nothing more than a superposition of multiple scenarios. The key transition that breaks the symmetry in many-vehicle situation must correspond to at least one of the 2-vehicle forced transitions.

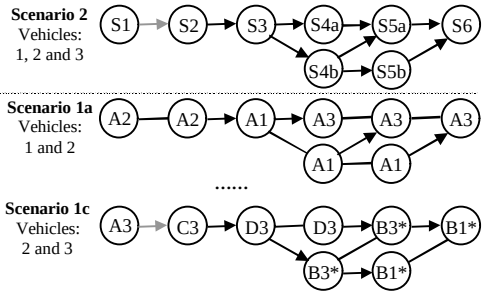

Figure 7. Three-vehicle transition diagram can be written as three separate two-vehicle transition diagrams.

Sometimes the interaction between two specific vehicles does not affect the multi-vehicle scenario considered for analysis. Therefore, by analyzing the interactions between vehicles whose actions affect the automata environment, we must be able to find a solution to the more complex situations (*e.g.*, the speed flag); the details are given in [6].

5. CONCLUDING REMARKS

Instead of trying to foresee all possible traffic situations, we define a mechanism based on the local sensor information in our non-model based approach. Definitions of the learning/sensor parameters determine the behavior of a vehicle, and they can be adjusted to guarantee safe operation. Our attempt to design an intelligent path planner extends, to some degree, to other levels of vehicle control. We have found that if a higher level of control/decision mechanism provides desired lane information, many local solutions may be extended to optimize overall traffic flow. The more global the information content of the decision mechanism, the more the vehicle can accomplish.

The method of evaluating possible environment state transitions based on automata environments enabled us to define additional decision mechanisms we called 'flags.' Speed and lane flags are used to solve the conflict situations arising from the multiple sensor module responses and vehicle interactions. Although our method of evaluating the state changes of the physical environment is based on the learning automata, similar methods can also be used with other decision mechanisms. By formal descriptions of the decision and control procedure, transition diagrams similar to those given in Section 4 can be created to analyze the highway situations.

6. REFERENCES

[1] Forbes, J., T. Huang, K. Kanazawa, and S. Russell, "The BATmobile: Towards a Bayesian Automated Taxi," *14th Intl. Joint Conf. on AI*, 1995.

[2] Lasky, T. L., and B. Ravani, "A review of Research Related to Automated Highway System (AHS)," Interim Rep. for FHWA, No. DTFH61-93-C-00189, UC Davis, Oct. 1993.

[3] K. S. Narendra and M. L. Thathachar, *Learning Automata: An Introduction*, Englewood Cliffs, NJ: Prentice Hall, 1989.

[4] Niehaus, A, and R. F. Stengel, "Probability-Based Decision Making for Automated Highway Systems," *IEEE Trans. on Veh. Tech.*, v. 43, no. 3, pp. 626-634, 1994.

[5] Sukthankar, R., *et al*, "Adaptive Intelligent Vehicle Modules for Tactical Driving," *13th Natl. Conf. on AI*, 1996.

[6] Ünsal, C., "Intelligent Navigation of Autonomous Vehicles in an AHS: Learning Methods and Interacting Vehicles Approach," Ph.D. Diss., Virginia Tech, Feb. 1997.

[7] Varaiya, P., "Smart Cars on Smart Roads: Problems of Control," *IEEE Trans. on Auto. Ctrl.*, v. 38., no. 2, Feb. 1993.

[8] Wheeler, R. M. Jr., and K. S. Narendra, "Learning Models for Decentralized Decision Making," *Automatica*, v. 21, 1985.