A FLEXIBLE MODEL FOR TRAFFIC SIMULATION AND TRAFFIC SIGNAL CONTROL OPTIMIZATION

TAN NGUYEN

This paper is submitted for COMP3740

Abstract. The aim of this work is to develop a flexible traffic model based on the recent development of traffic theory and apply the developed model to simulate traffic and signal control in order to identify the optimal signal control policy in several traffic scenarios. The paper starts with a review of the cell transmission model, the widely used model in traffic theory. Several impracticalities of the theory are pointed out and necessary modifications are offered in order to build a flexible computer based traffic model successfully. Finally, the developed model is used for simulation of different traffic scenarios with three common traffic signal control policies to identify their strengths and weaknesses, which is essential for optimizing traffic control and enables us to answer important questions like "Is the most widely used signal control policy (fixed time with offset) the most effective one?".

1. Introduction

Nowadays, as a consequence of the population boom, traffic congestion is a huge problem of every bigger city in the world. Its economical cost amounts to billions of dollars, not to mention its environmental impact. The apparent solution to this problem is to extend the infrastructure, but this solution is very expensive, time consuming, and is often subjected to environmental and social constraints. The other way around is to improve the traffic flow control, such that congestion is minimized. This second approach is much cheaper, faster, more efficient, and is not subjected to constraints that expansion of traffic infrastructure may have to cope with.

Traffic models play the crucial role in the study of optimization of traffic control. As planning can be only as good as the model, the traffic model is the number one factor that defines the success of a traffic control system. An automated traffic control system based on an unrealistic traffic model is a guaranteed disaster. Thus, more than ever, traffic models are being studied intensively by researchers all over the world.

It is very important for an implementation of a generic model to have declarative semantics rather than procedural semantics (e.g. Java code).

Date: May 17, 2011.

Key words and phrases. traffic flow, traffic modeling, traffic simulation, traffic signal control optimization, RDDL model.

This work was completed with the support of Dr. Scott Sanner.

This is because of the fact that models with declarative semantics can be exploited by other programs (planners) to analyze the transition semantics and build abstractions and/or apply different techniques like regression-planning or receding horizon control, which without declarative semantics would be impossible. The model presented in this paper uses RDDL (Relational Dynamic Influence Diagram Language) - see [2]. As its name may suggests, this is a declarative language for description of relationship of variables between transitions of the system. This language is extremely suitable for building traffic model, because the nature of traffic modeling is the encode of change of states of the traffic network when time advances from t to $t + 1$.

In this paper, the cell-transmission model (CTM) discussed by Daganzo in [1] and [2] will be studied. It is a well known macroscopic traffic model that gives relatively accurate prediction of macroscopic traffic flows (in consistent with kinematic wave theory) under all traffic condition. The model is reviewed in part 2 of this paper. Nonetheless, this model lacks some important factors when it come to practical implementation, perhaps due to some oversimplifications. Thus part 3 introduces some extension to the CTM in order to make it more flexible and practical for real traffic simulation. Part 4 discusses about an application of the developed model in traffic control optimization by designing an experiment of different traffic control policies applied to different traffic scenarios. Part 5 presents the overall conclusion of this paper, as well as suggestions for some possible future works.

2. Background

In traffic theory, when the level of detail is the main criterion, traffic models are classified into macroscopic models and microscopic models. A microscopic model traces every single vehicle in the system individually, so that during the simulation, all vehicles, together with their interactions, are simulated using the model. The sum of results of all simulated vehicles in the system generates an image or snapshot of the traffic situation. Input parameters of microscopic models are often very detail for every objects, e.g. for every section of roads it may requires speed limit, number of lanes, overtaking prohibitions etc.; for every vehicle it may requires origin, destination, desired velocity, acceleration and deceleration abilities, vehicle type, driving style (aggressiveness) etc. Because of its level of detail, microscopic models are computationally intensive and thus only suit simulation of a small section of the traffic network.

On the other hand, macroscopic models use aggregate variables to simulate traffic situation. Vehicles are no longer traced individually, they're rather considered collectively by traffic density (number of vehicles per kilometer), traffic flow (number of vehicles passing a road section per second), and average velocity of all vehicles. The first model of this kind was developed by Lighthill and Whitham [3] in 1955, where only traffic density was considered. Then the model is further extended by many others to be able to cope with shock waves and stop-and-go traffic in congested traffic situation. Macroscopic models, comparing to microscopic model, are less computationally intensive, thus allows a faster and larger simulation of the traffic network. They also have much fewer parameters than microscopic

models, which make them easier for fine tuning the traffic network control system. These advantages make macroscopic models much more suitable for optimization of traffic network.

The following section discusses the (macroscopic) CTM [2], and gives a quick overview of traffic control policies.

2.1. **The Cell-Transmission Model.**

2.1.1. *The Terminology.* In traffic theory, the traffic network is divided into cells. Each cell represents a section of the road. Thus there are fundamental physical quantities associated to a cell. These are traffic density, free flow velocity, traffic flow, and shockwave speed. Traffic density k (cars/m) is the number of cars per unit length. The maximum traffic density is called jam density, denoted by k_j. It is the maximum number of cars that can possibly be "compressed" into the given cell. Free flow velocity v (m/s) is the average speed of cars in the given cell when there's no traffic jam. It is often given by the speed limit of the road section represented by the given cell. Traffic flow q (cars/s) is the rate at which cars flow through a given point. Maximum traffic flow q_{max} is the maximum possible flow in accordance with the conditions of the cell. Shockwave speed w (m/s) is the speed with which disturbances propagate backward when traffic is congested.

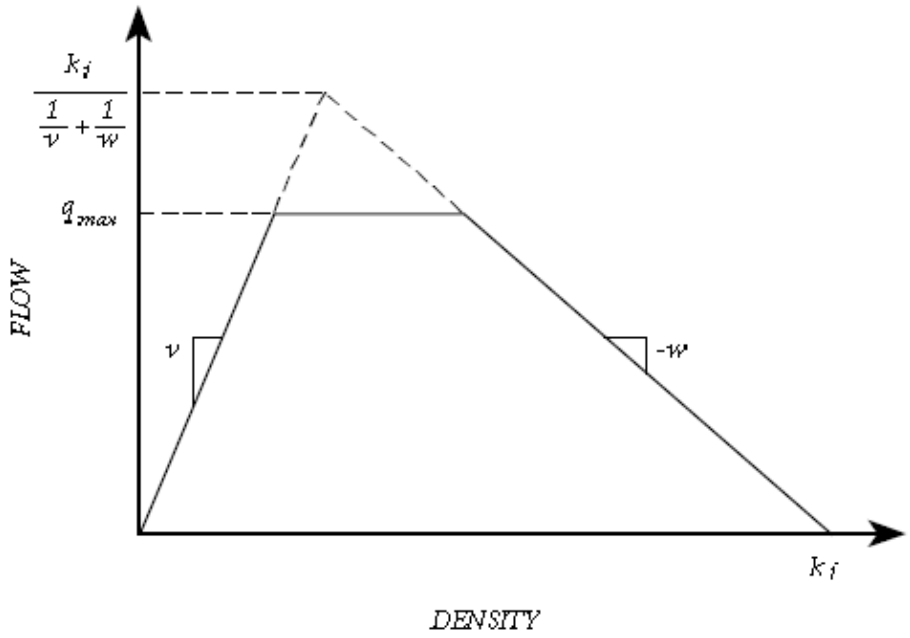

FIGURE 1. The equation of state of the cell-transmission model. [2]

The relationship between traffic flow q, density k and velocity v is naturally $q = v \cdot k$. However, q can't be greater than q_{max}, and in cases when there's traffic jam ahead, q can be only the "remaining" density times the shockwave speed. All these observations is expressed in graphical presentation given in Figure 1. When cars started flowing into the cell (from another

cell linked to this cell), the density k of the cell starts increasing, the flow $q = v \cdot k$ also increases constantly with the increase of k. Some time later, the flow q reaches its maximum q_{max} and can't be increased further. At this point more incoming cars only increase the density k of the cell. At a certain level of k, the cell is so dense, that incoming cars can enter the cell at a speed much slower than the free flow speed. It means the effective velocity is decreasing, causing a decreasing flow into the cell. At this stage the flow is $q = w(k_j - k)$. The shockwave speed w is the speed, with which traffic jam propagates backward. For example, at time t, the end of a 100m cell became stuck, and at time $t+5$, this event was propagated to the beginning of the cell (the beginning had became stuck), then the shockwave speed is $100/5 = 20$ m/s. When the cell's density reaches its maximum k_j, the flow is zero, there's no room for more cars to flow into the cell. Therefore the equation for relationship of all these basic variable is:

$$q = min\{vk,\, q_{max},\, w(k_j - k)\} \tag{1}$$

2.1.2. *CTM Network Representation.* The CTM assumes that the road is divided into homogeneous sections (cells) with lengths equal the distance traveled by free-flowing traffic in one clock interval. The traffic network is described as a graph of cells. The possible vehicle transfers between two cells are represented by a link between them. The topology of the detailed network is defined by specifying for each link k a "beginning" cell Bk and an "ending" cell Ek. For each cell i, Q_i is defined as the maximum number of vehicles that could enter cell i per unit time (one clock interval), N_i is defined as the maximum number of cars that can occupy cell i (maximum occupancy). The state of the network is then given by $\{n_i\}$: the set of number of vehicles contained in each cell of the network. As this is a macroscopic model, these numbers are not necessarily natural numbers, but rather real numbers. Note that comparing to the standard physical properties of a cell introduced in previous section, each cell in CTM is characterized by only two values: maximum throughput Q_i and maximum occupancy N_i. This significant simplification is due to the assumption above, that cells lengths equal the distance traveled by free-flowing traffic in one clock interval. By this assumption, in one clock interval, all cars in a cell Bk will move to the next cell Ek, assuming there's no traffic jam.

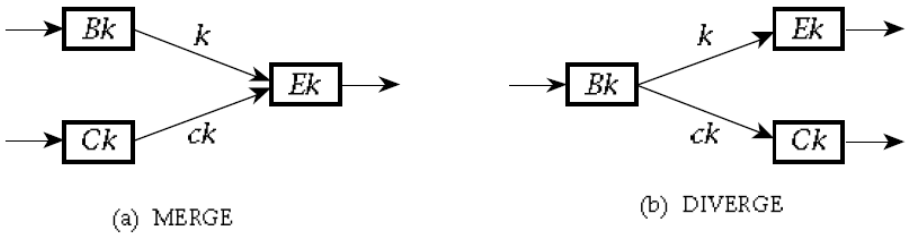

FIGURE 2. Representation of merge and diverge. [2]

The CTM further restricts the maximum number of arcs (links) entering and/or leaving a node (cell) to 3. Thus, cells can be classified into three types: "diverge" if only one link enters the cell but two leave it, "merge" if

two links enter and one leaves, and "ordinary" if one enters and one leaves (see Figure 2 above). By stacking up these basic elements we can represent any traffic network. For example an intersection can be represented by multiple diverges and merges connected together. This representation of the network greatly reduces the complexity of the network flow calculation, because it is now necessary to identify only three equations for the three linking types: ordinary, merge, and diverge.

2.1.3. *CTM Ordinary Links.* Ordinary link is the most easy one to calculate. Let $y_k(t)$ denote the flow on link k from clock tick t to clock tick $t+1$ between the beginning cell Bk and the ending cell Ek. Furthermore, for each cell I let $S_I(t)$ and $R_I(t)$ denote the sending and receiving capacity of that cell from clock tick t to clock tick $t+1$. Apparently, the sending capacity $S_I(t)$ can not exceed neither the maximum throughput Q_I nor the number of car in the given cell n_I. Thus, the equation for the sending capacity is:

$$S_I(t) = min\{Q_I, n_I\} \tag{2}$$

The receiving capacity also can not exceed the maximum throughput Q_I, and if maximum is not yet reached, it will equal a fraction of the "remaining" capacity of the cell $\delta_I[N_I - n_I]$, where $\delta_I = w_I/v_I$ denote the shockwave speed coefficient of cell I - note that in Equation (1) we have: $w(k_j - k) = (w/v)(N - n)$. Thus, the equation for the receiving capacity is:

$$R_I(t) = min\{Q_I, \delta_I[N_I - n_I]\} \tag{3}$$

Because the flow $y_k(t)$ can not exceed neither the sending capacity $S_{Bk}(t)$ nor the receiving capacity $R_{Ek}(t)$, its equation must be:

$$y_k(t) = min\{S_{Bk}(t),\, R_{Ek}(t)\} \tag{4}$$

2.1.4. *CTM Diverges.* Diverge can also be easily calculated if we know the turning percentages. Suppose we have a diverge as shown in Figure 2(b), and let $\beta_{Ek}(t)$ and $\beta_{Ck}(t)$ denote the proportions of $S_{Bk}(t)$ going each appropriate link. Apparently, $\beta_{Ek}(t) + \beta_{Ck}(t) = 1$, and:

$$\begin{aligned} y_k(t) &= min\{\beta_{Ek}(t)S_{Bk}(t),\, R_{Ek}(t)\} \quad \text{and} \\ y_{ck}(t) &= min\{\beta_{Ck}(t)S_{Bk}(t),\, R_{Ck}(t)\} \end{aligned} \tag{5}$$

Note that, in general, the percentages of left/right turn depends on the destinations of vehicles in the flow. A fixed assignment of this percentages like the above may not be very accurate. A better model is to determine these percentage dynamically from the "known" routes of the traffic flow, but that model goes well beyond the scope of this paper and thus will not be presented here.

2.1.5. *CTM Merges.* Merge is perhaps the most complicated one to calculate in CTM. Clearly, from Figure 2(a), the flows must satisfy these constraints:

$$y_k(t) \le S_{Bk}(t); \quad y_{ck}(t) \le S_{Ck}(t) \quad \text{and} \tag{6}$$

$$y_k(t) + y_{ck}(t) \le R_{Ek} \tag{7}$$

If $S_{Bk} + S_{Ck} \leq R_{Ek}$ then cell Ek receives all the flows from Bk and Ck. Thus we have:

$$y_k(t) = S_{Bk}(t) \quad \text{and} \quad y_{ck}(t) = S_{Ck}(t) \quad \text{if } R_{Ek} \geq S_{Bk} + S_{Ck} \tag{8}$$

In other cases we assume, that as long as the supply of vehicles from both approaches, $S_{Bk}(t)$ and $S_{Ck}(t)$, is not exhausted, a fraction p_k of the vehicles come from Bk and the remainder p_{ck} from Ck, where $p_k + p_{ck} = 1$. These "p" constants represent sort of priority of the appropriate flow.

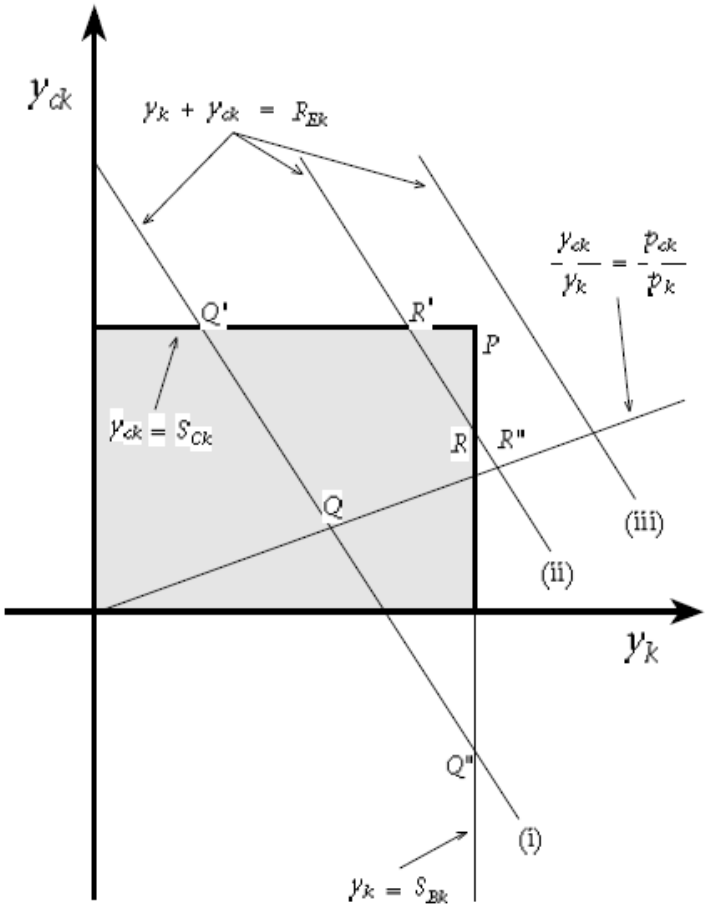

FIGURE 3. Graph of the merge equation constraints. [2]

Figure 3 shows the graph of possible flow for a merge. The shaded rectangle represents all solutions for Equation (6). The left half plane to the line $y_k + y_{ck} = R_{Ek}$ represent all possible solutions for Equation (7), three cases (i, ii, and iii) of possible position of that line are shown in the graph. The line $y_{ck}/yk = p_{ck}/p_k$ representing the flow proportion is also shown.

Case (iii) arises when $R_{Ek} \geq S_{Bk} + S_{Ck}$ and this is already solved by equation (9) (the solution is the coordinates of point P in the graph). In case (i) and (ii) the solution is the coordinates of point Q and R, because they maximize the possible flows, meanwhile satisfy all the given constraints.

We note that in both cases, the solution point is the middle point of the three intersection points of these four lines: $y_{ck} = S_{Ck}$, $y_k = S_{Bk}$, $y_{ck}/yk = p_{ck}/p_k$, and $y_k + y_{ck} = R_{Ek}$. Thus we have:

$$
\begin{aligned}
(9) \qquad & y_k(t) = mid\{S_{Bk}, R_{Ek} - S_{Ck}, p_k R_{Ek}\} \\
& y_{ck}(t) = mid\{S_{Ck}, R_{Ek} - S_{Bk}, p_{ck} R_{Ek}\} \\
& \text{if } R_{Ek} < S_{Bk} + S_{Ck}
\end{aligned}
$$

2.1.6. *CTM Network Update.* Once all flows have been calculated, it is easy to update the network: For each cell I, the new number of vehicles $n'_I(t+1)$ equals the old number of vehicles $n_I(t)$ plus sum of all flows into cell I, minus sum of all flows out of cell I. Note that it was mentioned before, that for a given cell, the maximum number of flows into a cell is two, likewise for flows out of a cell, and the total number of flows into and/or out of a cell is maximally three. Mathematically we write the network update function as:

$$
\begin{aligned}
(10) \qquad n'_I(t+1) = n_I(t) \quad & + sum\left\{y_{(B,I)} \,|\, linked(B,I)\right\} \\
& - sum\left\{y_{(I,E)} \,|\, linked(I,E)\right\}
\end{aligned}
$$

2.2. **Traffic Control Policies.** Control elements in a traffic network are traffic signals. Each traffic signal is often switched from one phase to the next phase, where each phase must strictly ensure the road safety (no traffic clash at any phase). For instance, traffic signal of an intersection of two one-way roads can have these predefined phases {GREEN WEST-EAST, YELLOW WEST-EAST, GREEN NORTH-SOUTH, YELLOW NORTH-SOUTH}, each of these phases corresponds to a configuration of traffic light, e.g. GREEN WEST-EAST means traffic from west to east is allowed, together with perhaps right-turn from west to south; traffic in north-south direction must not be allowed in this phase. A strategy for coordination of traffic signals in order to control the traffic network is called traffic control policy. There are different types of control policy, e.g. random policy, local adaptive policy, fixed time with offset policy. These policies are detailed below.

2.2.1. *Random Policy.* As its name suggests, the random control policy switches each traffic signal randomly and independently. The randomness here doesn't mean, that it is possible to have green light in all directions at the same time. Switching traffic signal means changing the signal from one phase to the next phase, and the randomness means the duration assigned to each phase is random. The parameters that define a random policy may be YELLOW-TIME (the fixed time for yellow light - there's no meaning to set yellow light time randomly!), MIN-GREEN-TIME (the minimum time for green light), MAX-GREEN-TIME (the maximum time for green light). The duration of a "green light" phase is set randomly between MIN-GREEN-TIME and MAX-GREEN-TIME. The min and max are there to ensure there's no extreme values, such as green light set for few second then switched to yellow.

2.2.2. *Fixed Time with Offset Policy.* This is the most common policy in practice. It works by assigning a fixed time signal switching for a critical (master) intersection, and offset time for other (slave) intersection. For example, let consider traffic signals A and B, with the distance from A to B is $d = 1000m$, and the free flow speed between A and B is $v = 20m/s$. Let assume A is a critical intersection and we want to coordinate the signal of B to maximize traffic throughput at A. Then the offset time between A and B should be set to $1000/20 = 50s$, so that when A is switched to green in time step t, B will be switched to green light in time $t + 50$. Meanwhile during time $t \to t+50$ traffic signal B can be set to handle traffic from other directions than from A.

2.2.3. *Local Adaptive Policy.* This policy uses local sensors to control the local traffic signal independently. It gives priority to the direction with more queuing traffic (that direction would receive longer green light time). There also should be parameters MIN-GREEN-TIME and MAX-GREEN-TIME to prevent extreme signal switching time, e.g. continuous switching in the situation that all directions are at their full capacity, or no switching at all in situation where there's to small flow in one direction.

3. Extended CTM

The CTM presented in the background part above is nice and clean. But in practice it is not convenient to have cell length equal the distance traveled by free-flowing traffic in one clock interval as it was assumed by the CTM. A cell, in practice, must have a defined length, so that in simple sections of the traffic network cells can be longer, and in crucial places cells can be shorter. The logic behind this is that longer cells means less computation time and shorter cells translate directly into accuracy. So in places like intersections, where fast update rate is required, cells would be defined with short length, and in other places like highways, where update rate is not so important, cells would be defined with long length to save computation time. This practical aspect of the problem calls for an extension of the CTM in order to have a somewhat more practical model for traffic simulation. This part suggests such extension to the CTM and shows how to implement the RDDL code for the extended model.

3.1. **Extension of The CTM.** As mentioned above, the extension must incorporate the cell length into the equations, so a cell now has these basic quantities: cell length l, jam density k, free flow velocity v, and actual number of vehicles in the cell n. All these quantities are real numbers. The task now is to reformulate all the CTM equations, but these are actually based on just two basic elements: the sending capacity and receiving capacity of the cell, so essentially it's enough to reformulate these two elements. After that, the rest of the CTM equations (based on those elements) are unchanged. We now start with the reformulation of the sending capacity, which indicates how many vehicles can flow out of the cell in one clock interval. Clearly, it must be the value of free flow velocity multiplied by the actual density of the cell, which is $v \times (n/l)$. But the sending capacity bounded by the

maximum flow, whose value is given by $v \times k$, so the new equation of the sending capacity can be written as:

$$S = min\{v\frac{n}{l}, vk\} \tag{11}$$

Assuming there's no traffic jam, then the cell will have maximum receiving capacity, which is $v \times k$. But if there is traffic jam, then the traffic flow should behave in accordance with the shockwave model, meaning that after sometime, the receiving capacity will be $w(k - (n/l))$ (see Equation (1)). Thus the equation of the receiving capacity of a cell can be written as follows:

$$R = min\{vk, w(n - \frac{n}{l})\} \tag{12}$$

Other equations in the CTM are based on the sending and receiving capacity, hence there's no need to reformulate them.

To the best of my knowledge, no one has provided these extensions before, but they are important for modeling real traffic scenarios and crucial for the empirical traffic control policy evaluation in the next part of this paper.

3.2. **RDDL Implementation of The Extended Model.** Relational Dynamic Influence Diagram Language (RDDL, see [4]) is a descriptive uniform language where states, actions, and observations are parameterized variables and the evolution of a fully or partially observed process is specified via functions over next state variables conditioned on current state and action variables. RDDL is extremely suitable language to describe the CTM. It would offers mechanism to implement :

- traffic signal, which is controlled independently by a concurrently executed action,
- real values for modeling variables and intermediate values,
- reward function as summing over all traffic cells (which change with each domain instance),
- transition function of next state based on the current state,

In RDDL, everything is a parameterized variable (fluent or non-fluent), the following lists the variables declaration part of the RDDL implementation of the extended CTM model, including traffic signal control.

```
// Types used in the model
types {
  cell : object;
  intersection : object;
  signal-phase : {@ALL-RED, @WEST-EAST, @ALL-RED2, @NORTH-SOUTH};
};

// Variables used in the model
pvariables {
  // Encoding of cell properties
  w(cell): {non-fluent, real, default = 40.0}; // shock-wave speed
  v(cell): {non-fluent, real, default = 20.0}; // free-flow speed
  l(cell): {non-fluent, real, default = 80.0}; // cell length
  k(cell): {non-fluent, real, default = 0.3};  // jam density
```

```
  // Encoding of traffic network topology
  NEXT(cell, cell): {non-fluent, bool, default = false};
  FIRST(cell): {non-fluent, bool, default = false};
  LAST(cell): {non-fluent, bool, default = false};

  // Encoding of merges and diverges
  MERGE-CELL(cell): {non-fluent, bool, default = false};
  DIVERGE-CELL(cell): {non-fluent, bool, default = false};
  p(cell, cell): {non-fluent, real, default = 0.0}; //percentages

  // Encoding of traffic signals
  signal(intersection): {action-fluent, signal-phase,
                                          default = @ALL-RED};
  FLOWS-TO-INTERSECTION(cell, intersection, signal-phase):
                            {non-fluent, bool, default = false};

  // Actual number of vehicles in cell
  n(cell): {state-fluent, real, default = 0.0};

  // Intermediate vars: sending/receiving capacity & flow
  send(cell) : {interm-fluent, real, level = 1};
  recv(cell) : {interm-fluent, real, level = 1};
  flow(cell, cell) : {interm-fluent, real, level = 2};
};
```

The first part of the code declares types used in the model. Cell and intersection are object types, signal-phase is an enumerate type. Different problems describe different traffic networks, hence they have different number of cells, intersections, etc. Because of this reason the list of objects are declared in the problem instance of each problem.

The non fluent variables are predefined characteristics of the traffic network. Their value, if different from the default value, are defined in the declaration of the problem instance. Variables *w, v, l, k* define the fundamental parameters of the cell: shock-wave speed, free-flow velocity, length, and jam density. Variables *NEXT* defines all links in the traffic network (does cell c1 flow to cell c2?). The variable *FIRST* marks a cell as a charging cell, which initially has a large number of vehicles defined in the problem instance definition (other cells has no vehicles at the beginning). The task of this cell is to pump vehicles into the traffic model at a given rate (defined by the density value of the *NEXT* cell to the *FIRST* cell). The variable *LAST* defines a discharging cell, which is a cell with a very large value of density, it imitates an unlimited parking to store all vehicles that have gone out of the network. The sum of vehicles in *LAST* cells is the reward and we would like to maximize it. Variable *MERGE-CELL, DIVERGE-CELL, p* define the all merges and diverges, *MERGE-CELL / DIVERGE-CELL* mark the cell as a merge / diverge; *p* stores the priority percentages for merge's inflows and turn probabilities for diverge's outflows. For the encoding of traffic signal, variable *FLOWS-TO-INTERSECTION* determines if a cell flows into an

intersection. The action fluent variable, which is controlled in every time step by a control policy, *signal* determines the phase of the signal. If a cell flows into an intersection then it is subjected to the signal control policy of that intersection.

The state fluent variables in RDDL encode states of the problem. These are the variables that need to be updated from the current state to next state, which is a transition of the model. The state fluent variables are updated using intermediate fluent variables. In this problem we only have one state fluent variable: n - number of vehicles in each cell, and three intermediate variables *send, recv, flow* representing sending capacity, receiving capacity and flow, just like in the CTM model. The RDDL code for the update process is listed below for illustration.

```
// Define the conditional deterministic function for each next
// state variable in terms of previous state and action
cdfs {
  send(?c) = if (v(?c) * n(?c)/l(?c)< v(?c) * k(?c)) then
                v(?c) * n(?c)/l(?c)
             else
                v(?c) * k(?c);

  recv(?c) = if (v(?c) * k(?c) < w(?c) * (k(?c) - (n(?c)/l(?c)))) then
                v(?c) * k(?c)
             else
                w(?c) * (k(?c) - (n(?c)/l(?c)));

  // Apply many nested if ... then ... else to calculate flows
  // in accordance with the extended CTM model
  flow(?c1, ?c2) = if ...

  // Finally, update the next state of the network
  n'(?c) = n(?c) + [sum_{?c1 : cell} flow(?c1, ?c) * NEXT(?c1, ?c)]
                 - [sum_{?c2 : cell} flow(?c, ?c2) * NEXT(?c, ?c2)];
};

// General conditions (constraints) that must always hold
state-action-constraints {
  // ensure two cells flow into a merge cell
  forall_{?c: cell} [MERGE-CELL(?c)
                     => ((sum_{?c1 : cell} NEXT(?c1, ?c)) == 2)];
  // ensure a diverge cell flows into two other cells
  forall_{?c: cell} [DIVERGE-CELL(?c)
                     => ((sum_{?c2 : cell} NEXT(?c, ?c2)) == 2)];
};

// Reward is the sum of number of vehicle that have arrived
reward = sum_{?c : cell} n(?c) * LAST(?c);
```

Figures 4 and 5 show the visualization of the RDDL model in various instances. The visualization is written in Java, which read states from the

RDDL model and draw the appropriate traffic network diagram. Note that the cell color is interpolated from white to red in accordance with the density level of the cell (from zero to maximum density). The cells with green color are either a charging cell or a discharging cell. Each cell have a cyan triangle indicating the existence of traffic movement and its direction. Traffic signal is indicated by either an green arrow showing the allowed direction (green light direction), or a yellow circle representing that all directions is not allowed.

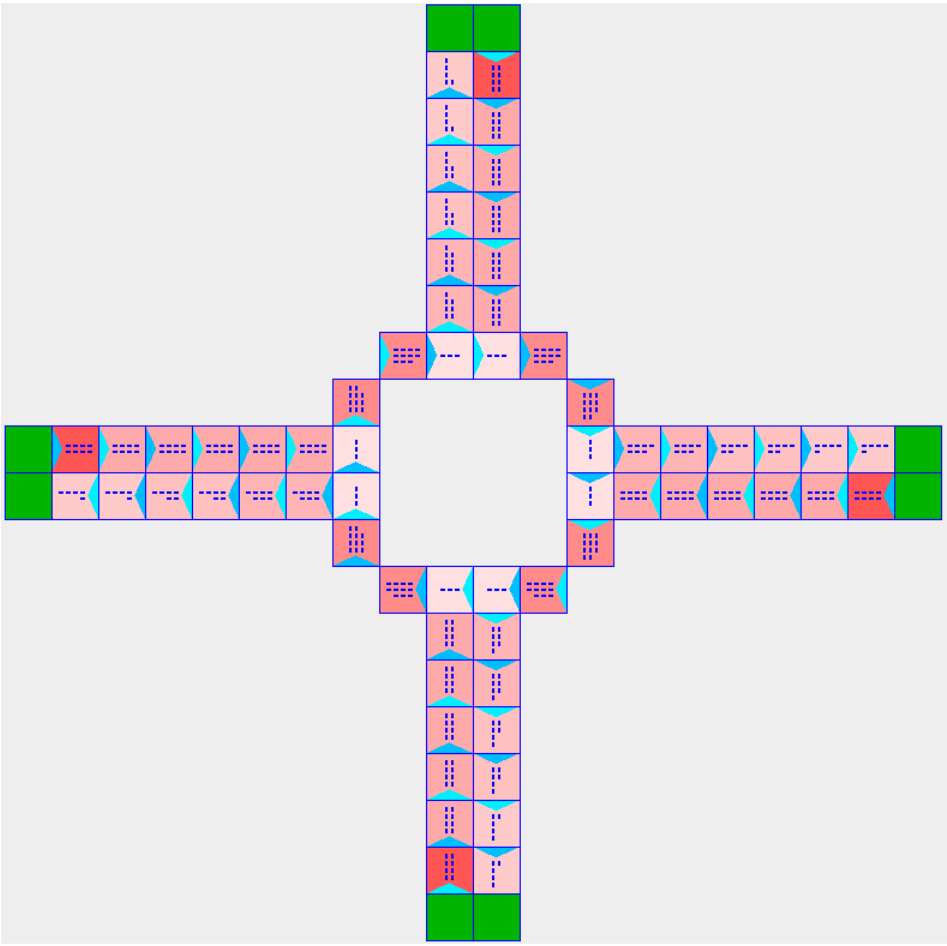

FIGURE 4. Visualization of a roundabout

4. Policy Evaluation

4.1. **Experiment Description.** This part represents an empirical traffic control policy evaluation using the extended CTM. A 3x3 grid of one-way roads is used to simulate a part of a city (see Figure 5). There are traffic signal at each of the 9 intersections of the grid. The distance between adjacent intersection is 400m. Nine traffic scenarios are considered as follows:

- High traffic in all directions
- High traffic west-east and medium traffic north-south

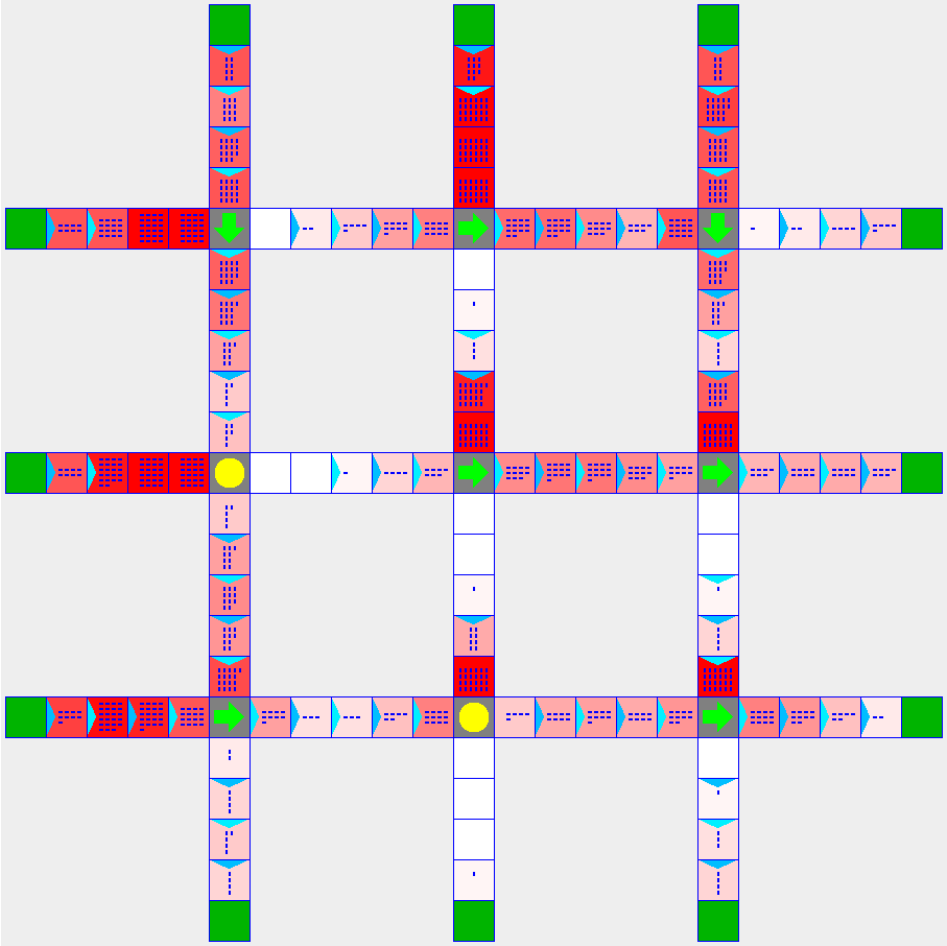

FIGURE 5. Visualization of a grid of 3x3 one-way roads with 9 intersections including traffic signal at every intersection

- High traffic west-east and low traffic north-south
- Medium traffic west-east and high traffic north-south
- Medium traffic in all directions
- Medium traffic west-east and low traffic north-south
- Low traffic west-east and high traffic north-south
- Low traffic west-east and medium traffic north-south
- Low traffic in all directions

The three traffic control policies described in the background part are tested against the above 9 traffic scenarios in order to assess their effectiveness in each case. Each control policy is set with optimal parameters for the given traffic grid. A script is then used to run each control policy with each traffic scenario and store the results to an output file. The result of each test is the number of vehicles that managed to go through the traffic network (that is equivalent to the number of vehicles arrived to $LAST$ cells). For every test, the number of time steps is set to 2000 (approximately half an hour) to ensure the run-time is long enough, so that the policy fully exposed its potential.

4.2. **Experiment Result.** The following tables list the results of the experiment:

Table 1. Random policy test result

W-E / N-S	HIGH	MEDIUM	LOW
HIGH	17877	17956	12756
MEDIUM	17755	17731	12767
LOW	12821	12924	7618

Table 2. Fixed time with offset policy test result

W-E / N-S	HIGH	MEDIUM	LOW
HIGH	20159	19794	13911
MEDIUM	19775	19411	13527
LOW	13872	13872	7624

Table 3. Local adaptive policy test result

W-E / N-S	HIGH	MEDIUM	LOW
HIGH	18826	18714	19815
MEDIUM	18744	18588	15248
LOW	19852	15231	7621

4.3. **Experiment Evaluation.** From the result tables above, it can be easily seen that the random policy has the worst performance, with an average of approximately 10% lower performance than the other two policies. On the other hand, it was quite difficult to set the optimal parameters for the fixed time with offset policy to optimize traffic flow in both directions, but the symmetry of the results shows that the parameters were set correctly (otherwise there would be a significant difference between the result of e.g. west-east high, north-south low traffic versus west-east low, north-south high traffic).

The interesting findings is the comparison of performance between fixed time with offset policy and the local adaptive policy. While the fixed time with offset policy has about 5% higher performance in all scenarios where the the traffic in both directions is from medium to high, the local adaptive policy really shows its advantage in cases where traffic is low in one direction and above low level in the other direction with performance about 20% higher than that of fixed time with offset policy (note that if traffic is low in both directions then the performance is the same regardless of what policy is used). This is perhaps an important finding, because in practice, many intersections would have the same characteristics suitable for adaptive control policy (one direction with low traffic, the other with high traffic), and we have shown that by replacing the commonly used fixed time with offset policy by local adaptive policy we would have a boost of about 20% traffic throughput in these intersection.

Table 4 summarizes the suitability of each type of policy for each traffic scenario. FT means fixed time with offset policy, AD means local adaptive policy, ANY means any policy is suitable

TABLE 4. Suitability of traffic control policy for each traffic scenario.

W-E / N-S	HIGH	MEDIUM	LOW
HIGH	FT	FT	LA
MEDIUM	FT	FT	LA
LOW	LA	LA	ANY

5. Conclusion and Future Work

This paper has extended the CTM model in order to produce a practical traffic model in RDDL. The model is then used in an empirical evaluation of several traffic control policies, which has successfully pointed out the suitability of each type of control policy for certain traffic scenarios. It showed, that the common fixed time with offset control policy is not the best policy in all cases, and a suitable combination of fixed time with offset control policy with local adaptive control policy may significantly boost the performance of the traffic network.

Due to a very limited time scope of the project, this paper hasn't considered and/or explored several problems to a greater details. For example: the implementation of a planner to optimize parameters of control policies, the model of known routes instead of the turn probability for diverges, the optimization of the model for better computational performance etc. These perhaps cane be subjects of work in the future.

References

1. Daganzo C.F., *The cell-transmission model: A simple dynamic representation of highway traffic.* Department of Civil Engineering and Institute of Transportation Studies, University of California, Berkeley CA 94720 (1993)
2. Daganzo C.F., *The cell transmission model: Network traffic.* Department of Civil Engineering and Institute of Transportation Studies, University of California, Berkeley CA 94720 (1994)
3. Lighthill M. J. and Whitham G.B., *On kinematic waves II. A theory of traffic flow on long crowded roads.* (1955).
4. Sanner S., *Relational Dynamic Influence Diagram Language (RDDL): Language Description.* NICTA, Australia (2011)

RESEARCH SCHOOL OF COMPUTER SCIENCE, COLLEGE OF ENGINEERING & COMPUTER SCIENCE, AUSTRALIAN NATIONAL UNIVERSITY

E-mail address: u4702662@anu.edu.au