

# Dynamic Distributed Lanes: Motion Planning for Multiple Autonomous Vehicles

Rahul Kala\* and Kevin Warwick

*School of Systems Engineering, University of Reading, Whiteknights,*

*Reading, Berkshire, United Kingdom*

rkala001@gmail.com, k.warwick@reading.ac.uk

<http://rkala.in/>, <http://www.kevinwarwick.com/>

Tele: +44-7424752843

\* *Corresponding Author*

**Citation:** R. Kala, K. Warwick (2014) Dynamic Distributed Lanes: Motion Planning for Multiple Autonomous Vehicles. *Applied Intelligence*, 41(1): 260-281.

**Final Version Available At:** <http://link.springer.com/article/10.1007%2Fs10489-014-0517-1>

## Abstract

Unorganized traffic is a generalized form of travel wherein vehicles do not adhere to any predefined lanes and can travel in-between lanes. Such travel is visible in a number of countries e.g. India, wherein it enables a higher traffic bandwidth, more overtaking and more efficient travel. These advantages are visible when the vehicles vary considerably in size and speed, in the absence of which the predefined lanes are near-optimal. Motion planning for multiple autonomous vehicles in unorganized traffic deals with deciding on the manner in which every vehicle travels, ensuring no collision either with each other or with static obstacles. In this paper the notion of predefined lanes is generalized to model unorganized travel for the purpose of planning vehicles travel. A uniform cost search is used for finding the optimal motion strategy of a vehicle, amidst the known travel plans of the other vehicles. The aim is to maximize the separation between the vehicles and static obstacles. The search is responsible for defining an optimal lane distribution among vehicles in the planning scenario. Clothoid curves are used for maintaining a lane or changing lanes. Experiments are performed by simulation over a set of challenging scenarios with a complex grid of obstacles. Additionally behaviours of overtaking, waiting for a vehicle to cross and following another vehicle are exhibited.

**Keywords:** *autonomous vehicles, robotics, graph search, planning, multi-robot path planning, multi-robotic systems*

## 1. Introduction

The advantages of autonomous vehicles over the human-driven vehicles include safety, efficiency and personal comfort [1]. Autonomous vehicles, connected via an inter-vehicle communication framework [2-3], may ‘talk’ and inform each other about the road scenario. They may further communicate with the road infrastructure [4], a processing unit fixed with the roads. The vehicles may hence make collaborative plans which are better than human or machine-based planning without communication wherein one has to ‘guess’ the intent of the other driver. The advantages include traffic jam avoidance [5], collaborative obstacle avoidance [6], etc. Attributed to the advantages of autonomous vehicles, it will become increasingly possible to have autonomous vehicle only traffic in the future for large sections of city/motorway traffic, once such autonomous vehicles are commercially available. This is hence a common assumption in the transportation literature.

Most countries follow organized traffic where the road width is divided into a number of lanes and vehicles are asked to drive within a lane, changing lanes if necessary. The task of planning is primarily to decide, at every instance of time, the lane and speed of travel [7-8]. On the contrary some countries like India stick to unorganized traffic wherein a vehicle can travel anywhere on the road and sticking to a particular lane is not a requirement [9-11]. The width of a lane, in organized traffic, is kept just larger than the maximum width of a vehicle. Vehicles larger than a lane’s width are not allowed, while much smaller vehicles (excluding motorbikes) are very rare. If traffic is under the occupancy of vehicles of varying sizes, from very small to large, the unutilized width of lanes in occupancy with smaller vehicles would add up to a significant amount, enough to fit in a lot more vehicles. This is the main reason why unorganized Indian traffic shows a higher bandwidth as

compared to that possible with organized traffic (restricted to the number of lanes), where the vehicles vary in sizes from cycles, motorbikes, 3-wheeled auto rickshaws, cars to big trucks.

Further, big differences in speed between vehicles mean that every overtaking manoeuvre must be completed as soon as possible, with the cooperation of the other vehicles involved. In unorganized traffic this is possible as soon as space is available, while in organized traffic one would have to wait for the overtaking lane to be clear. Unorganized Indian traffic shows multiple simultaneous overtakes, wherein cars overtake 3-wheeled auto rickshaws, which again overtake cycles; while motorbikes overtake or get overtaken depending upon the situation/driver. Each such overtake is useful as it is 'painful' to follow a slower vehicle at its speed. Further, if small obstacles lie on all lanes of travel, all the lanes are rendered un-navigable, while it may still be possible to navigate in between the obstacles, which is what many human drivers, even in organized traffic, tend to do.

Sticking to lanes is a good traffic rule if the majority of the vehicles in the traffic are wide enough to occupy the entire lane and the standard deviation in speed is small. However, as the size and speed diversity increases, the limitations of this traffic rule become increasingly apparent, which explains why some countries stick to unorganized traffic. Unorganized traffic is more generalized in nature and is capable of displaying a richer set of traffic behaviours which are restricted in organized traffic due to the lane-sticking traffic rule. Motion planning for unorganized traffic is important in order to get autonomous vehicles into countries following unorganized traffic. Further, fully autonomous vehicles not requiring a human inside may have speed and size designed to best cater for business requirements. These vehicles may bring large diversity to the traffic landscape, questioning the validity of lanes in the presently organized countries.

Interesting research has been done for planning a vehicle in organized traffic operating within lanes, which is not applicable for unorganized traffic. Furda and Vlacic [12] modelled the problem as a deterministic state automata problem based on raw sensor data, using which the authors exhibited robotic behaviours of lane keeping, maintaining a safe distance from other vehicles and avoiding any potential collisions. Multi-Criterion Decision Making was used for strategy selection. Reveliotis and Roszkowska [13] meanwhile modelled the system as a resource allocation problem with lanes as virtual resources, which the vehicles attempted to acquire. Conflict resolution strategies for allocation prohibited collisions. Schubert et al. [14-15] fed the distances from between the lane markings and the vehicles into an automaton to compute the optimal lane of travel.

Some attempts have been made for motion planning without necessarily the assumption of lanes. Kala and Warwick [16] studied the problem of planning and coordination of vehicles by coarsely formulation all possible plans of a vehicle (assuming the other vehicles travel straight), from which the best plan was optimized using an elastic strip. Kuwata et al. [17] used Rapidly-exploring Random Trees for the planning of a single vehicle using a biased sampling technique for obstacle avoidance. Chu et al. [18] constructed a number of trajectories for obstacle avoidance using a single manoeuvre, out of which the best was used. Gehrig and Stein [19] used an elastic band attached to a leader vehicle to model the motion of the follower vehicles. These approaches can however all only exhibit simple behaviours of vehicle following, overtaking or obstacle avoidance; but cannot display the strong coordination strategies possible with the availability of communication.

The problem of planning in unorganized traffic is closer to the problem of multi-robot motion planning, for which a number of interesting approaches exist. The difference between wide maps used in mobile robotics, as opposed to the narrowly bounded roads in the case of vehicles does though question the validity of many approaches. Other differences include the notion of overtaking and vehicle following, terms which are largely baseless in robotics; the absence of a predefined goal in vehicles; the continuous emergence of vehicles in a road, unlike in mobile robotics where robot positions are initially known and any addition causes re-planning.

Centralized approaches [20] are non-implementable in real time, due to which planning needs to be performed in a decentralized manner [21-23]. Pure decentralized approaches cannot though model strong cooperation between the vehicles which is only possible with centralized approaches. Hence the use of a decentralized approach must ensure that the resultant system can showcase all types of desirable behaviours between a pool of vehicles, e.g. in prioritized approaches [24] lower priority vehicles always lose out, which can lead to overtaking becoming unfeasible due to a lack of cooperation.

Graph search based methods are widely used for robotic planning. However a centralized graph search approach for multi-robots is clearly computationally too expensive. For a single robot approaches rely on the use of hierarchies for computational speedup. However in a road based system, it is impossible to make any decision (to overtake or not, side to avoid an obstacle, etc.) at a lower resolution map, since all such behaviours take place with small separations which must be precisely known. Kala et al. [25] presented an iterative graph search

algorithm, where the map was represented at multiple resolutions. The resolution around the path was increased along with iterations to get better paths. Lu et al. [26] also used an iterative graph search, employing the Lifelong Planning A\* algorithm for a dynamic environment. Chand and Carnegie [27] decomposed the problem at two levels, using the A\* algorithm for both levels. A finer A\* algorithm computed the connectivity cost between coarser level nodes. Cowlagi and Tsiotras [28] studied the problem of different kinematics for different robots, and modelled a graph with multiple costs at the edges depending upon the kinematics and state. Model predictive control was then used for local trajectory tracking.

This paper is motivated by an earlier work by the authors [29] where a 4 layer planning algorithm was proposed consisting of route selection, an obstacle avoidance strategy, distribution of available (obstacle-free) road width amongst the vehicles, and trajectory generation. But a limitation of the work was that the different layers worked at different abstractions and the decisions of any particular layer were made based on abstract knowledge of the other vehicles available for that layer. However, as most decisions relating to overtaking, obstacle avoidance, and other manoeuvres take place with small separations, precise information is needed for optimal decision making. Hence instead of dividing the algorithm into hierarchies, here the intention is to stick to the finest hierarchy and use heuristics to limit the state expansions, thereby forming a completely different method of making graph searches faster.

The key contributions of this paper are: (a) A generalized notion of lanes is defined. (b) The generalized notion is used for planning and coordination of multiple vehicles. (c) A coordination technique is designed which uses the concepts of decentralized coordination for iteratively planning different vehicles but empowers a vehicle to pass other vehicles. The coordination is hence better in terms of optimality and completeness than purely decentralized approaches, while being somewhat computationally expensive in the worst cases. (d) The concept of one vehicle waiting for another vehicle coming from the other direction is introduced, when there may be space enough for only one vehicle to pass. (e) Heuristics are used for pruning the expansions of states, which result in a significant computational efficiency while leading to a slight loss of optimality.

Section 2 deals with the generalized notion of lanes. Section 3 presents the algorithm framework for a single vehicle, which is extended for the presence of multiple vehicles in section 4. Results are given in section 5. Discussion is included in section 6 and section 7 contains the concluding remarks.

## 2. Generalized Notion of Lanes

Inspired by organized traffic, we first address the question of what constitutes lanes in unorganized traffic, which is used as the basis of the planning algorithm. In other words the concept of lanes is generalized here from organized to unorganized traffic.

A (generalized) lane is defined as the portion of road (without obstacles) occupied exclusively by a vehicle for a specific duration of time. The vehicle may be found at any location inside a portion of road with a guarantee that it will not collide with any other vehicle. Some characteristics of lanes, which must be considered to make an optimal travel plan, are:

- (i) *Distributed* – The number of lanes and their widths may change in different segments of the road. Consider a thin slice of road, which is segmented so as to only include obstacle-free regions. The number and width of lanes change as the slice is traversed along the length of the road. Both the number and widths of lanes at the road slice depend on the width of the road, presence of obstacles, demand of the road slice at any instance of time, and widths of vehicles demanding the road slice.
- (ii) *Dynamic* – The lanes change with time as vehicles pass by. The number and widths of lanes at a road slice change with time. For every vehicle the lane changes with time as the vehicle discovers more vehicles it may possibly overtake, completes an overtake, is overtaken, decides to travel aside some vehicle, and similar situations.
- (iii) *Single vehicle* – No two vehicles may occupy a lane side-by-side.
- (iv) *Variable Width* – Every lane has a different width that depends upon the width of the vehicle that uses it and the total available width (excluding obstacles).

## 3. Planning for a Single Vehicle

### 3.1 Problem and Vehicle Design

The first problem we consider is planning a single vehicle. A road segment (part of the entire road), characterized by its two boundaries, is given; which may consist of any number/type of static obstacles. Let  $R_i$  be the vehicle to be planned, initially located at position  $(x^s_i, y^s_i)$  with orientation  $\theta^s_i$ ; and having current speed  $v^s_i$  ( $\leq v_{max}^i$ , the maximum permissible speed). Further consider that the vehicle is a rectangular grid of known size  $len_i \times wid_i$ . Non-rectangular vehicles are expanded to the nearest rectangular shape or handled as per the bounding box approach [30-31]. For algorithmic purposes, the X-axis is taken as the longitudinal axis, along the direction of the road; and the Y-axis is taken as the lateral axis. Considering the possibility of irregular width roads, the Y-axis coordinate value is taken as a ratio to the current road width [29].

Let the free configuration space of the vehicle be given by  $\zeta_{static}^{free}$ . The initial configuration of the vehicle  $S$  is known. The algorithm computes the trajectory  $\tau_i(t)$  for the duration  $0 \leq t \leq T_i$ . Here  $T_i$  is the time for the vehicle to reach the planned end point. The objective of the algorithm is to make the vehicle travel as far as possible in the road segment (or *maximize*  $\tau_i(T_i)[X]$ ). For trajectories that reach equally as far, the one that takes the shortest time is selected (or *minimize*  $T_i$ ). If the vehicle travels further, there is less chance of it being struck in an obstacle framework with no way out other than reversing, if there is no single collision free trajectory which can overcome all obstacles as per the limited view of the current road segment being planned.

The planning algorithm is a uniform cost search [32-33] using which all possible expansions are made. A state or node is represented by the vector  $L_i \langle x_i, y_i, \theta_i, v_i, t, \tau_i, lane_i \rangle$  where  $(x_i, y_i)$  denotes position,  $\theta_i$  denotes orientation,  $v_i$  denotes speed (constant) and  $t$  is the time of arrival at the state,  $\tau_i$  denotes trajectory from the source that leads to the state, and  $lane_i$  denotes the generalized lane from the source to the state. To make discussions simple, in this text we use the term state to refer to both a state of the configuration space (consisting of  $\langle x_i, y_i, \theta_i \rangle$  only) as well as a node of the graph search. The cost of a state  $L_i$  is taken to be its time, or  $L_i[t]$ . Once all the nodes are expanded, the best trajectory is selected. The general algorithm is given in Algorithm 1.

#### Algorithm 1: Uniform Cost Search for a single vehicle

##### Plan( $R_i$ )

```
Source  $\leftarrow S_i$ 
Source[ $\tau_i$ ]  $\leftarrow$  NULL, Source[lane $_i$ ]  $\leftarrow S_i$ 
Q  $\leftarrow$  Source, Q $_2$   $\leftarrow$  NULL
while Q  $\neq$  NULL
    Q  $\leftarrow$  Sort(Q)
    P  $\leftarrow$  Expand(Q(1))
    Q $_2$   $\leftarrow$  Q $_2 \cup$  Q(1)
    Q  $\leftarrow$  (Q - Q(1))  $\cup$  P
end Q
qp  $\leftarrow$  best plan in Q $_2$ 
return qp[ $\tau_i$ ], qp[lane $_i$ ]
```

### 3.2 State Reduction

For a rich set of possible actions, the graph search is computationally expensive and cannot be used. Hence we use heuristic means to carefully select the actions or states to generate in expansion. Two hypotheses are used regarding the motion strategy of the vehicle. (a) Every vehicle attempts to move in the road such that its relative position (on the lateral axis) remains the same. This is naturally seen while we drive on roads when no steering is performed while travelling on a straight road. Required turns are made in a manner so as to keep oneself at approximately the same relative position compared to others. (b) The vehicle always attempts to move such that its separation from obstacles or road boundaries is as large as possible [34]. However the attempt to maximize the separation is limited to a value of  $sm$ . This is again visible in everyday driving where in the presence of any obstacle we attempt to align the vehicle, maximizing the separation from the obstacle.

Expansion of a state  $L_i \langle x_i, y_i, \theta_i, v_i, t, \tau_i, lane_i \rangle$  of the graph search is done in a manner that the vehicle moves forward by a magnitude of  $\Delta(v_i)$  in the X axis. The value in the Y axis is chosen by the above mentioned

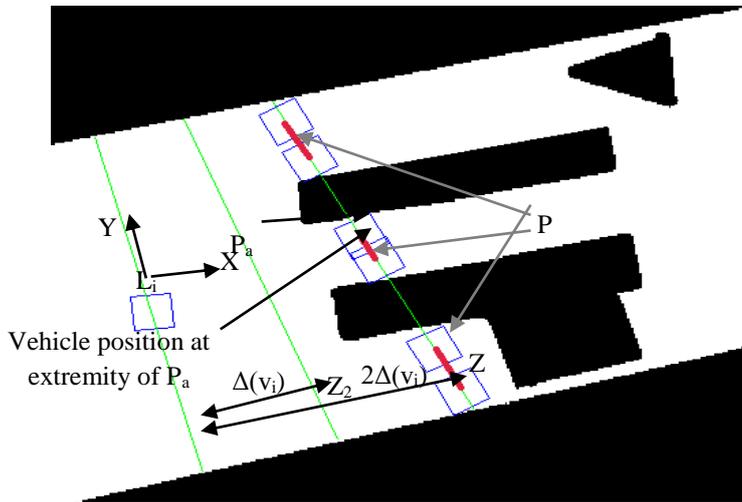
hypotheses. The choice of new position for the vehicle needs to be done in a manner so as to maximize the separation of the vehicle in both the lateral (X) and longitudinal (Y) axis. Lateral separation is maintained by the hypothesis while longitudinal separation is maintained by  $\Delta(v_i)$ . If an obstacle happens to lie just ahead of the vehicle it may not be able to further continue its journey without collision. The aim is to keep a minimal distance of  $\Delta(v_i)$  from the obstacle in front.  $\Delta(v_i)$  (equation (1)) must be large enough so that the vehicle can steer itself a significant amount laterally, while travelling longitudinally on the road. Higher speeds would naturally need larger longitudinal distance (along the X axis) to steer across the Y axis.

$$\Delta(v_i) = c \cdot v_i \quad (1)$$

Here  $c$  is a constant called as the *longitudinal separation constant*.

The first step in expansion is to analyse the obstacles in the region. The analysis of the obstacles is done at a distance of  $2\Delta(v_i)$  from the current position (in order to guarantee a longitudinal separation of  $\Delta(v_i)$  from the expanded state). We consider the *obstacle analysis line Z* as the Y axis at a longitudinal distance of  $2\Delta(v_i)$  from  $x_i$  (Figure 1). Vehicles mostly align themselves along the road, and hence while crossing the line Z, the ideal vehicle's position should be perpendicular to Z. We find valid states  $P$  given by equation (2).

$$P = \bigcup_l \langle x_i, y_i, \theta_i, v_i, t \rangle : (x_i, y_i) \in Z, \theta_i \perp Z, l \in \zeta_{static}^{free} \quad (2)$$



**Figure 1: Obstacle analysis**

We select minimal states from the set  $P$  as the expanded states. The line  $Z$  may cross obstacles, which means all possible states  $P$  would be disjoint into smaller sets  $P_a$  ( $P_a \subset P, P_a \cap P_b = \emptyset \forall P_a, P_b, P_a \neq P_b$ ) (Figure 1). We reduce the states to be formed by expansion to one per disjoint set. Figure 1 also shows the importance of selecting states from each set, as two of the three sets are later discovered to be sub-optimal (or blocked).

### 3.3 State Selection

The selection of a state to be expanded ( $l$ ) in the set  $P_a$  is based on the set hypothesis. Let  $d(l_3)$  be the distance of  $l_3$  lying on  $Z$  measured from the road boundary. Let  $s_1$  and  $s_2$  be the points in  $Z$  which have the least and the highest value of  $d(\cdot)$  such that the line  $s_1$  to  $s_2$  is collision free and includes  $P_a$  (Figure 2). Going by the set hypothesis, disregarding the obstacles, the vehicle may attempt to keep itself in the same lateral position on the road which gives its preferred position  $pr$  such that  $pr[Y] = y$  (or  $L_i[Y]$ ) through which we may write equation (3)

$$d(pr) = y \cdot rl \quad (3)$$

Here  $rl$  is the road width at  $Z$ .  $pr$  may though be such that a vehicle placed at  $pr$  is not feasible and within  $P_a$  or may not keep a distance of  $sm$  from obstacles or road boundaries, which if introduced gives the preferred position  $l$  as in equation (2).

$$d(l) = \begin{cases} d(pr) & d(s_1) + sm + wid_i / 2 \leq d(pr) \leq d(s_2) - sm - wid_i / 2 & (a) \\ d(s_1) + sm + wid_i / 2 & d(pr) < d(s_1) + sm + wid_i / 2 \wedge d(s_2) - d(s_1) \geq 2sm + wid_i & (b) \\ d(s_2) - sm - wid_i / 2 & d(pr) > d(s_2) - sm - wid_i / 2 \wedge d(s_2) - d(s_1) \geq 2sm + wid_i & (c) \\ (d(s_1) + d(s_2)) / 2 & otherwise & (d) \end{cases} \quad (4)$$

Equation 4(a-c) are applied when it is possible for the vehicle to have a margin of  $sm$  on both sides. Equation 4(a) states the condition when the vehicle as per  $d(pr)$  already maintains a wide margin at both sides. Equation 4(b) and 4(c) apply for conditions when the vehicle has a wide margin at one side but not on the other side, and needs to be moved so that a margin is produced on both sides. Equation 4(d) is an attempt to maximize the separation as much as possible by placing the vehicle in the middle of the segment. Various cases that arise in selection of  $l$  are shown in Figure 3. For details refer to equation 4(a-d).

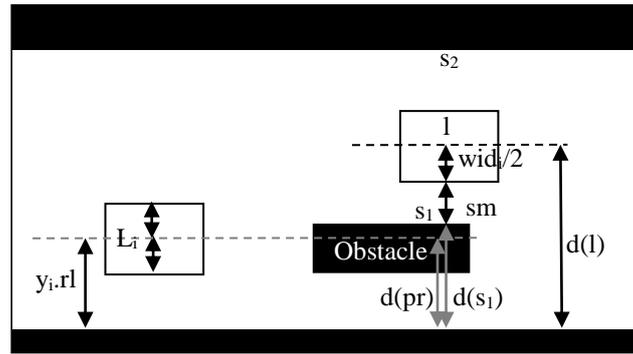


Figure 2: Computing state for expansion

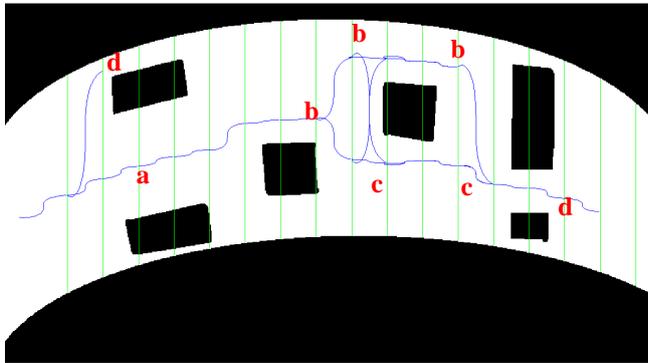


Figure 3: Various cases in selection of state for expansion

However the state  $l$  lies on  $Z$  which may be quite close longitudinally to an obstacle, even though lateral separation has been maximized. The purpose was to find a valid state at a longitudinal distance of  $\Delta(v_i)$ , say  $l_2$ . Let  $Z_2$  (*vehicle placement line*) be the  $Y$  axis at a distance of  $\Delta(v_i)$ .  $l_2$  (equation 5) is selected as a state such that the vehicle can steer itself and orient itself parallel to the road attaining the correct relative (lateral) position of  $l[Y]$  when it crosses the line  $Z_2$ . It may then later travel parallel to the road to state  $l$  (Figure 4).

$$l_2 \langle x_i, y_i, \theta_i, v_i, t \rangle : (x_i, y_i) \in Z_2, y_i = l[Y], \theta_i \perp Z_2 \quad (5)$$

The *obstacle analysis line* ( $Z$ ) keeps the algorithm informed about the obstacles it may face in the future and the vehicle must receive its correct orientation well in advance. The motion takes place only till the *vehicle placement line* ( $Z_2$ ), after which (in the next expansion)  $Z$  is further moved to keep the vehicle informed about

still further obstacles. In this way  $Z$  acts as a forerunner for the vehicle, while  $Z_2$  acts as the line up to which the vehicle can confidently travel.

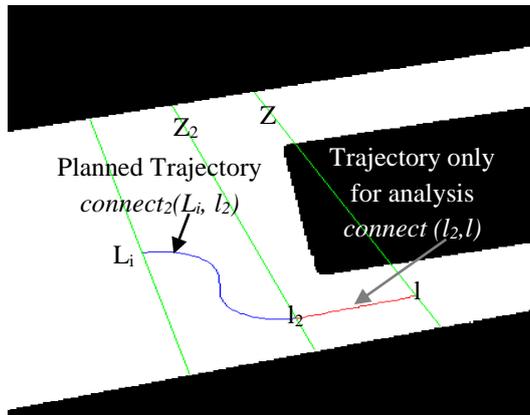


Figure 4: Object analysis and state expansion

### 3.4 Curve generation

The point  $L_i$  is connected to  $l_2$  using clothoid curves [35-36] denoted by  $connect_2(L_i, l_2)$ . The purpose of clothoids is to generate a curve that makes the vehicle correct its angle of orientation, change its lateral position in the road and generate a curve for the vehicle to traverse on a curved road.  $l_2$  is connected to  $l$  by a curve parallel to the road denoted by  $connect(l_2, l)$  (Figure 4). The feasibility of the two curves is checked such that all intermediate points belong to  $\zeta_{dynamic}^{free}$ . A small local search around  $l$  is performed to assess if either of the curves is infeasible.

Infeasibility may be due to a *small obstacle problem* shown in Figure 5. The line  $Z$  completely misses out citing the small obstacle and hence the computations are erroneous. In such a case, feasibility may only be returned by  $Z$  discovering the small obstacle for which  $\Delta(v_i)$  is avoided by a reduction of speed. Reduction of speed by a discrete amount at a time instant is assumed to be possible as in instantaneous models [37-38].

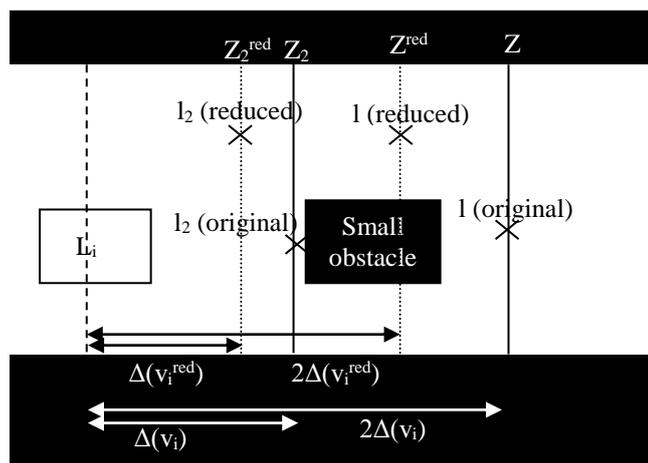


Figure 5: Small Obstacle Problem

If the curves are feasible  $l_2$  becomes the expanded state with parent  $L_i$ , and is added in the processing queue of the uniform cost search. The region in which  $l$  can be moved in  $Z$  and correspondingly  $l_2$  can be moved in  $Z_2$  without causing a collision (under the threshold of  $sm$  at both sides from  $l$ ) constitutes a vehicle's lane at  $l_2$ . The lane definition for tentative motion from  $L_i$  to  $l_2$  can be visualized by connecting the lane definition at  $L_i$  to the lane definition at  $l_2$  along the longitudinal axis. The points  $L_i$  and  $l_2$  characterize the lane, and these are stored for

algorithmic purposes. The expansion is given as a pseudo-code in Algorithm 2. The graph generated for a synthetic problem is shown in Figure 6.

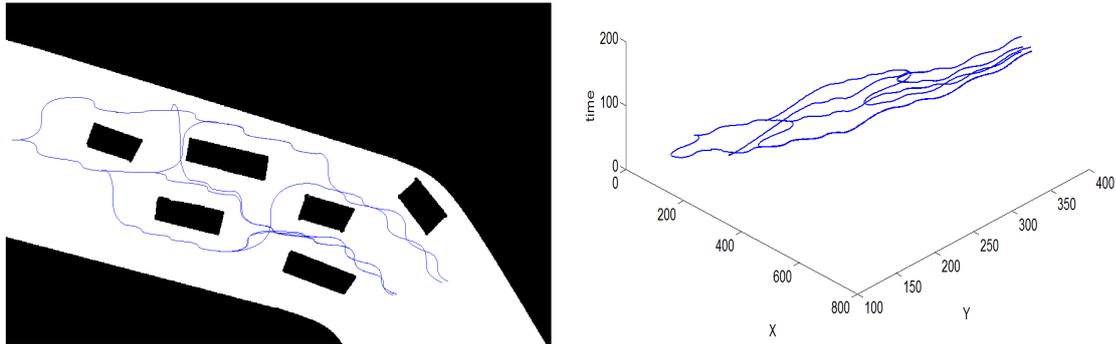


Figure 6: Graph generated for a sample problem (a) for given map (b) showing time axis

**Algorithm 2: Expansion for a single vehicle**

**Expand( $L_i$ )**

$A \leftarrow \text{NULL}$

for  $v_i \leftarrow v_{\max}^i$  to 1 in small steps

    Calculate  $P$  using (2)

    for all  $P_a \in P$

        calculate  $l, l_2$  for  $P_a$  using (4) and (5)

$[l, l_2] \leftarrow$  apply small deviations to  $l$  till  $\text{connect}_2(L_i, l_2) \in \zeta_{\text{dynamic}}^{\text{free}} \wedge \text{connect}(l_2, l) \in$

$\zeta_{\text{dynamic}}^{\text{free}}$ , break  $P_a$  if still infeasible

$l_2[\tau_i] \leftarrow L_i[\tau_i] \cup \text{connect}_2(L_i, l_2)$

$l_2[\text{lane}_i] \leftarrow L_i[\text{lane}_i] \cup l_2$

$l_2[t] \leftarrow L_i[t] + \|\text{connect}_2(L_i, l_2)\|/v_i$

$A \leftarrow A \cup l_2$

    end  $P_a$

    if  $A \neq \text{NULL}$ , return  $A$

end  $v_i$

return  $\text{NULL}$

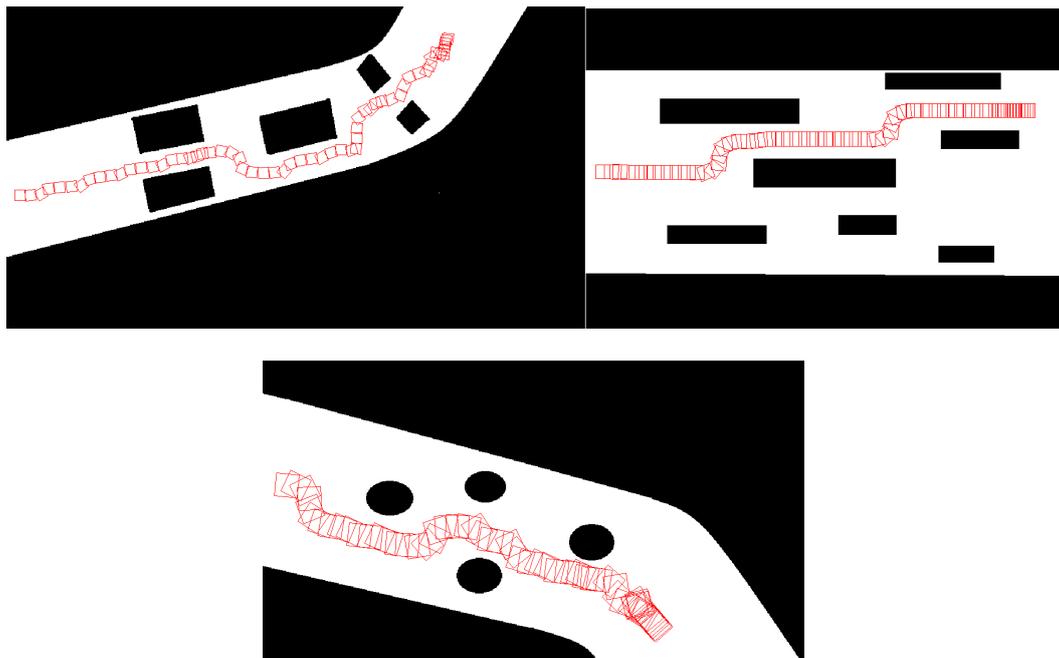


Figure 7: Experimental results with single vehicle

### 3.5 Experimental Results

The algorithm was developed in MATLAB with the map being given as a bmp image file. The aim of testing was to uncover the potential of the algorithm to compute optimal paths in a simple to complex grid of obstacles. The first scenario (Figure 7(a)) was created to test the ability of the vehicle to enter narrow regions. Multiple such placements at small distances stress a clever steering strategy which is more difficult if it is to be done at high speeds. Another scenario was made to test the ability of the vehicle to pass through a complex network of obstacles (Figure 7(b)). It may be seen that the vehicle could do well in the scenario making only a few steering attempts. In the third scenario (Figure 7(c)) some simple obstacles were placed at random places. It may be seen that the vehicle could detect all the obstacles and pass by them, either by a reduction of speed or by local optimization.

## 4. Planning with Multiple Vehicles

### 4.1 Problem and Algorithm Design

The problem consists of  $N$  vehicles, each vehicle  $R_i$  with its own time of emergence  $e_i$  in the planning scenario. The source  $S_i$  which is the state of the vehicle at  $e_i$  is known, but only after vehicle emergence. Each vehicle is planned upon its emergence. When  $R_i$  arrives, the trajectories  $\tau_j$  of vehicles  $R_j$  are already known for  $e_j < e_i$ . The aim is to generate a plan  $\tau_i$  for  $R_i$ . The plan  $\tau_i$  is called admissible in the planning scenario  $\tau$  if no collision occurs between any two vehicles or static obstacles. If  $R_i$  fails to generate an admissible plan, the plans of all vehicles are nullified and re-planning is done. The order of planning may be altered and any combination that results in a valid plan may be selected.

We make use of a uniform cost search. The first task is obstacle analysis using the *obstacle analysis line* ( $Z$ ), which gives a set of states  $P$  in disjoint sets of  $P_a$ , disjoint by static obstacles. This is used to select the expanded state. Here however we need to take into account the other vehicles as well. The general algorithm is the same as Algorithm 1.

$R_i$  may not be able to construct a feasible plan without moving the other vehicles by modifying their plans.  $R_i$  is allowed to alter the lanes of the other vehicles, which correspondingly changes their trajectories. The lane of any vehicle  $R_j$  is in the form of a set of points  $lane_j[u]$  while the trajectory  $\tau_j$  is a result of connecting these points by a smooth curve. The points  $lane_j[u]$  act as controls available with  $R_i$ , which may alter any of these to produce a different lane for  $R_j$  and hence change the trajectory.

Every graph node (state) maintains trajectories  $\tau$  detailing all alterations made to the trajectories of the vehicles in the path from the source to the state in the graph expansion; and  $lane$  detailing all alterations made to the lanes of vehicles. Hence every child takes the plan (trajectories and lanes) specified by the parent, makes modification to it as per its desires, and stores the resultant plan which becomes the plan specification given to all its children which may modify it further.

Consider any graph node or state  $L_i < x_i, y_i, v_i, \theta_i, t, \tau, lane >$  which needs to be expanded using the search technique. We formulate three expansion strategies and any one or more of the strategies may be used for the expansion. These are *free state expansion*, *vehicle following* and *wait for vehicle*. Each of these is discussed in the following sub-sections.

### 4.2 Free State Expansion

The *free state expansion* strategy attempts to place every vehicle laterally with  $R_i$  moving with the highest possible speed. In the absence of other vehicles, this strategy is similar to the expansion strategy discussed in section 3. The expansion might lead the vehicle to overtake any slow moving vehicle if it comes on its way. Hence the strategy may also be termed as an *overtaking strategy*.

The problem is the expansion of state  $L_i$  for the valid states  $P_a$  returned by static obstacle analysis. Here we do not select a single position in  $P_a$  for the vehicle  $R_i$ , that was done in planning for a single vehicle. Rather we divide  $P_a$  (specifically, complete *obstacle analysis line* segment  $Z$  bounded by road boundary or obstacle from both ends) between the vehicles which plan to move parallel to  $R_i$  in its motion till  $Z$ , and must hence occupy distinct lanes. Let there be  $n$  such vehicles. The task is the selection of states  $l^1, l^2, \dots, l^n$  where  $l^i$  denotes the state

of  $R_j$  on crossing  $Z$ . The task is hence twofold, first to find the vehicles that need to occupy distinct lanes, and second to identify the lane for each of these vehicles ( $l$  is known as the lane for  $R_j$  at the particular state).

Imagine a thick slice of road with a vehicle travelling through the slice amidst other vehicles moving under their own plans. If there seems to be a likely collision between any two vehicles, one would like to alter the travel plans of the two vehicles such that they lie in different lanes. Carrying out and completing the alteration of the plan of the new vehicle can span the considered road slice and hence the slice may need to be widened. If the changed plan results in a different collision, one would like to include the new colliding vehicle, and assign all 3 vehicles different lanes. Alterations may result in further widening of the slice. The altered plan may further cause more collisions, in which case the participating vehicles are added. The addition is done till all vehicles can harmoniously cross the road slice. Once the vehicles are identified, the (obstacle free) road slice can be divided between the vehicles, such that each occupies an independent lane across the width of the road slice. This example illustrates the two tasks, to identify the vehicles requiring independent lanes, and to divide the road slice into different lanes between these vehicles.

#### 4.2.1 Computing the number of lanes required

The first task is calculating the vehicles that need separate lanes while  $R_i$  crosses  $Z$ . As per the hypotheses we need to maximize the lateral separation of  $R_i$  from any other vehicle or obstacle, subjected to a maximum of  $sm$ , while longitudinal separation may preferably be  $\Delta(v_i)$ . As per the free state expansion strategy, all vehicles that as per their plan lie at a longitudinal separation of less than  $\Delta(v_i)$  need an independent lane. This however excludes the *following a vehicle behaviour*, when the vehicle ahead moves with a greater or equal speed. Giving an independent lane assures that no collision happens as long as the vehicles stick to their allocated lanes, irrespective of what longitudinal separation they maintain. Suppose we select a vehicle  $R_j$  needing a separate lane, which would be required to alter its plan. Modification of the plan of  $R_j$  might make the longitudinal separation between  $R_j$  and any other vehicle  $R_k$  less than the required  $\Delta(v_j)$ , threatening a collision between  $R_j$  and  $R_k$ . In this manner we keep adding vehicles until we reach a point where modification of lanes does not affect the other vehicles.

Let set  $H$  denote the vehicles (including  $R_i$ ) requiring independent lanes. Initially the set  $H$  contains only  $R_i$ . Suppose at time  $t$ ,  $R_j$  as per its planned trajectory is at position  $\tau_j(t)$ . Let  $S(S_a, S_b)$  denote the segment of road under consideration for computing lanes, where  $S_a$  is the start position in the  $X$  axis and  $S_b$  is the end position in the  $X$  axis. Within the region  $S$  vehicles must occupy independent lanes, unless a vehicle follows another. Since we are studying the prospective motion of  $R_i$  from its current position  $x_i$  to its final position at a longitudinal separation of  $2\Delta(v_i)$ , the initial value of the segment  $S$  is  $(x_i - len_i/2, x_i + 2\Delta(v_i) + len_i/2)$ , accounting for the entire coverage of  $R_i$  in this motion (Figure 8).

The sets  $H$  and  $S$  are iteratively grown until no further vehicle qualifies to be added to  $H$ . At any time, every  $R_k$  is checked such that motion of any  $R_j$  in  $H$  within  $S$  may result in a collision between  $R_j$  and  $R_k$ . This separation check is only done at points denoting lanes of  $R_j$ . If the separation is less than  $\Delta(v_j)$ , there is a possible collision. In such a case,  $R_k$  is added to  $H$  and accordingly  $S$  is modified to account for the segment  $S_2$  that is the region affected by  $R_k$  being given an independent lane. The modified  $S$  is given by  $S \cup S_2$ . The procedure is repeated until no vehicle in the planning scenario qualifies to be added in  $H$ . The two sub-problems that arise are: calculate  $S_2$  if some  $R_j$  is given an independent lane, and calculate  $S$  if some  $R_j$  in  $H$  threatens some  $R_k$ , requiring it to have an independent lane, each of which is dealt with one after the other. The general algorithm is given in Algorithm 3.

#### Algorithm 3: Getting number of vehicles requiring independent lanes

##### getLaneVehicles( $L_i$ )

$A \leftarrow R_j: e_j < e_i, H \leftarrow R_i$

$S(S_a, S_b) \leftarrow (x_i - len_i/2, x_i + 2\Delta(v_i) + len_i/2)$

do

  for  $R_j \in A - H$

    if  $\tau_j(t) \otimes R_j \cap (S_a, S_b + d) \neq \emptyset$

$S_2 \leftarrow (lane_j[a+1] - len_j/2, lane_j[b+2] + len_j/2)$

$S(S_a, S_b) \leftarrow S \cup S_2$

$H \leftarrow H \cup R_j, A \leftarrow A - R_j$

      break  $R_j$

    end if

  end  $R_j$

while no change in H  
return H

Getting into a new lane means  $R_j$  needs to steer to get to the new lane, and later steer back to the original lane (Figure 8); the positions when both these changes happen needs to be calculated. Both turns require a distance of  $\Delta(v_j)$  along the X axis. New lanes are brought into effect immediately, or  $R_j$  is immediately asked to go to the new lane. Suppose  $R_j$  at time  $t$  has left the point  $lane_j[a]$  and moves towards the point  $lane_j[a+1]$  (or its motion has been confirmed) as per its planned trajectory. In such a case we cannot alter the motion of the vehicle from  $lane_j[a]$  to  $lane_j[a+1]$ . Modification of  $lane_j[a+1]$  would mean a different trajectory connecting  $lane_j[a]$  to  $lane_j[a+1]$ . The part of trajectory that vehicle has already travelled from  $lane_j[a]$  to  $lane_j[a+1]$  may not be the same as the one desired with an altered  $lane_j[a+1]$ . However we can certainly modify lanes  $lane_j[a+2]$  and onwards (equation (6)).

$$a: lane_j[a][T] \leq t \wedge lane_j[a+1][T] > t \quad (6)$$

Here  $lane_j[a][T]$  denotes the time  $R_j$  arrives at the lane, while  $lane_j[a]$  denotes the position of the lane.

At the time of expansion of  $L_i$ , the algorithm assumes that  $R_i$  suddenly disappears after it achieves the expanded state at the *vehicle placement line*  $Z_2$ , while no collision would be recorded in moving from  $L_i$  to  $Z$ . The subsequent graph expansions cater for the latter motion. Hence after  $R_i$  attains the expanded state, other vehicles must return to their original lanes.  $R_i$  ends its journey at a longitudinal distance  $\Delta(v_i)$  from  $x_i$ , after which vehicles must maintain an additional longitudinal safety distance of  $\Delta(v_i)$  before going to their original lanes (Figure 8). Vehicles may be driving inbound or outbound. Considering both cases,  $R_j$  can return to its original lane after it is completely out of the region of considered motion of  $R_i$  (considering lengths of both vehicles contribute to their longitudinal occupancy) at  $lane_j[b]$  (equation (7)). For  $lane_j[b+1]$  onwards,  $R_j$  has normal lanes.

$$b: lane_j[b] \otimes R_j[Y] \cap (x_i - len_i/2, x_i + 2\Delta(v_i) + len_i/2) \neq \phi, lane_j[b][T] > lane_j[a][T] \quad (7)$$

Here  $lane_j[b] \otimes R_j[Y]$  denotes the longitudinal coverage of  $R_j$  placed at  $lane_j[b]$ .

No vehicle must lie at a separation of  $\Delta(v_j)$  from  $lane_j[b+1]$ , which is the location of  $lane_j[b+2]$ . Hence  $S_2$  is given by  $(lane_j[a+1] - len_j/2, lane_j[b+2] + len_j/2)$ . The notations are illustrated in Figure 8. An assumption here is that after the vehicle  $R_i$  reaches the line  $Z$ , all vehicles may more or less simultaneously travel back to their original lanes. This may lead to collisions when one vehicle readily moves back to its original lane, while another vehicle takes a little time to start moving back. These are handled by optimization during trajectory generation.

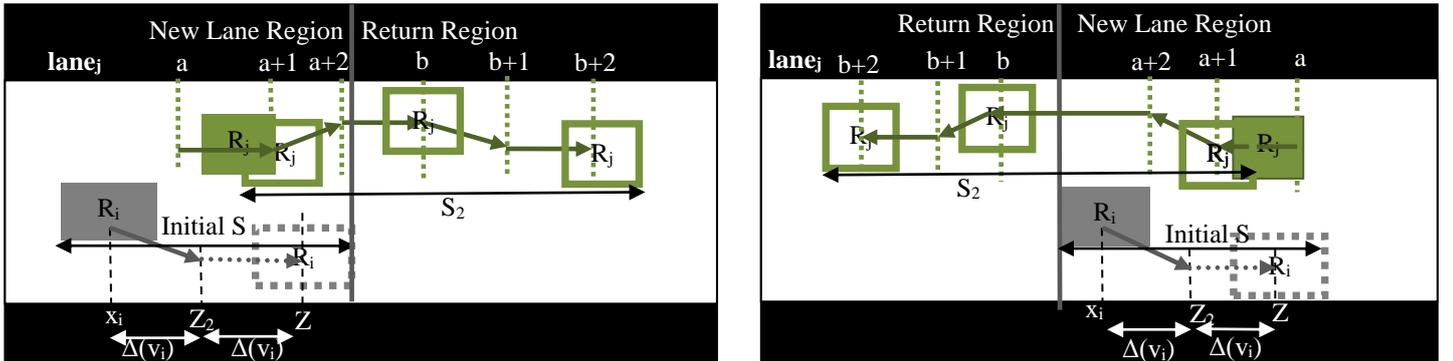
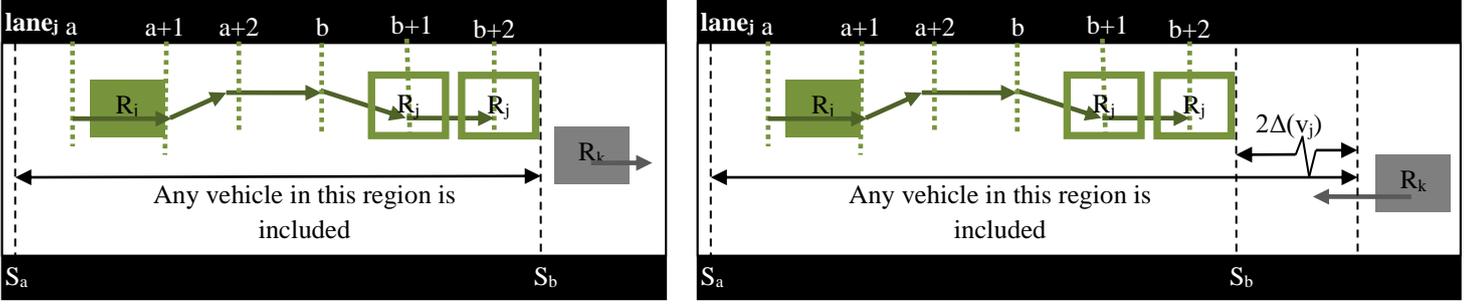


Figure 8: Calculating region of independent lanes (a) same direction of travel (b) opposite direction of travel

The last task is to check if the changed dynamics of any  $R_j$  cause a potential threat with any general  $R_k$ , indicating the set  $H$  to be grown to include  $R_k$ . Since the changes in lanes of  $R_j$  are not yet computed (section 4.2.2), the time when  $R_j$  crosses  $lane_j[b]$  is unknown. Due to this a threat to any  $R_k$  cannot be determined. This issue may be handled by heuristics. Knowing the positions of the vehicle at time  $t$  we aim to decide the separation of the vehicles. We divide the decision into two cases. These are whether  $R_j$  and  $R_k$  are travelling in the same directions (both inbound or outbound) or different directions (one inbound and the other outbound). As a general rule  $R_k$  needs to be included in  $H$  if any part of  $R_k$ , while at  $\tau_k(t)$ , lies within  $(S_a, S_b+d)$ .

In the case when vehicles are travelling in the same direction,  $d$  is taken to be 0. The worst case is when  $R_j$  reaches its normal lane at  $lane_j[b+1]$  and  $R_k$  has not moved much making the separation between  $R_j$  and  $R_k$  as small as possible. The inclusion of extra distance in computing  $S_2$  for  $R_j$  ensures this distance is wide enough. Any motion of  $R_k$  would make the separation more than the minimum amount of  $\Delta(v_j)$ . The notations are shown in Figure 9(a). In the case when vehicles are travelling in different directions,  $d$  is taken to be  $2\Delta(v_k)$ . Here since  $R_j$  races towards  $R_k$  as  $R_k$  travels, the distances get shorter very quickly. The smallest distance would be recorded when  $R_j$  returns to its normal lane. Since we do not know the time required for the same, and hence the distance travelled by  $R_k$  in the process, we calculate an approximate value assuming longitudinal travel for both vehicles (minimum of  $2\Delta(v_k)$ ), as the maximum distance that can be covered by  $R_k$  while  $R_j$  turns to a modified lane and again back to its normal lane. The notations are shown in Figure 9(b).



**Figure 9: Decision regarding inclusion of  $R_k$  in  $H$  (a) same direction of travel (b) opposite direction of travel**

#### 4.2.2 Lane distribution

The task is division of the entire available width of the free road into lanes between the vehicles demanding the same. We use the same notations as in planning for a single vehicle. Let  $s_1$  and  $s_2$  be the points in  $Z$  which have the least and highest value of  $d(\cdot)$  (distance from boundary) such that the line  $s_1$  to  $s_2$  is obstacle free and includes  $P_a$ . Let  $rl$  be the road width at  $Z$ . The preferred position of any  $R_j$  in  $H$  is given by equation (8).

$$d(pr_j) = \begin{cases} lane_j[a][Y].rl & j \neq i \\ y.rl & j = i \end{cases} \quad (8)$$

The different  $pr_j$  may be infeasible or without maximum separations. We sort the vehicles in  $H$  as per their  $d(pr_j)$  values.  $pr_i$  may not be within the segment, which is corrected by giving it the highest or lowest possible value. Working with any  $R_j$  to create maximum separation is similar to the approach used in the single vehicle case, with the addition that  $R_j$  might move other vehicles to create extra space for itself. Let  $left(pr_j)$  (equation (9)) be the point to which  $R_j$  placed at  $pr_j$  may be moved leftward without colliding with any other vehicle, road boundary or obstacle. The value of  $left(pr_j)$  is the distance of that point from the boundary. Similarly for right we have  $right(pr_j)$  given by equation (10). Notations are shown in Figure 10.

$$left(pr_j) = \begin{cases} d(pr_{j+1}) - wid(pr_{j+1})/2 & j < size(H) \\ s_2 & j = size(H) \end{cases} \quad (9)$$

$$right(pr_j) = \begin{cases} d(pr_{j-1}) + wid(pr_{j-1})/2 & j > 1 \\ s_1 & j = 1 \end{cases} \quad (10)$$

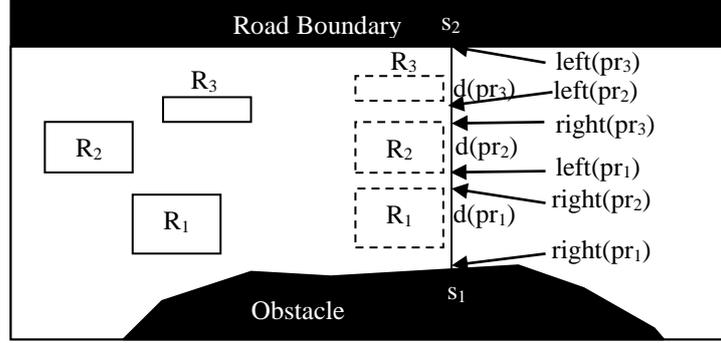


Figure 10: Distribution of lanes

The first case that arises is the possibility of placing  $pr_j$  such that no movement of other vehicles is required and it would maintain a distance of  $sm$  from both its ends. Equation (11) gives the precondition and equation (12) gives the placement.

$$C = left(pr_j) - right(pr_j) - wid_j \geq 2sm \quad (11)$$

$$d(l^j) = \begin{cases} d(pr_j) & right(pr_j) + sm + wid_j / 2 \leq d(pr_j) \leq left(pr_j) - sm - wid_j / 2 \\ right(pr_j) + sm + wid_j / 2 & d(pr_j) < left(pr_j) - sm - wid_j / 2 \wedge C \\ left(pr_j) - sm - wid_j / 2 & d(pr_j) > right(pr_j) + sm + wid_j / 2 \wedge C \end{cases} \quad (12)$$

Equation (12) in structure represents equation (4). However in case condition  $C$  is not met, the task is to move the vehicles to have more space for accommodating  $R_j$ . A vehicle is only moved if the vehicle already has a distance excess of  $sm$  from both its ends. This is the cooperation shown by the vehicles wherein they allow  $R_j$  to have sufficient separation by effectively lessening their own excessive separation. The total separation needed by  $R_j$  is  $s$  (equation (13)). The spare separation that  $R_k$  has and is ready to offer is given by equation (14).

$$s = \max(2sm - (left(pr_j) - right(pr_j) - wid(pr_j)), 0) \quad (13)$$

$$s_k = \max(left(pr_k) - right(pr_k) - wid(pr_k) - 2sm, 0) \quad (14)$$

Vehicles  $R_k$ , one by one on either side of  $R_j$  are moved, till the required separation  $s$  is created. Vehicles on the left of  $R_j$  are moved towards the left and those on the right of  $R_j$  are moved towards the right to get the necessary separation, that is movements are centred along  $R_j$ . Notations are shown in Figure 10.

In case, for any vehicle, the required separation is unavailable, the total separation available  $((s_2 - s_1) - \sum_j wid(pr_j))$  is evenly distributed between the vehicles. If all vehicles get sufficient space without overlapping each other, the planned distribution may be used for the planned movement of the vehicles. The complete algorithm is given by Algorithm 4.

#### Algorithm 4: Division of the road into lanes

**getLanePositions(H, L<sub>i</sub>)**

calculate  $pr$  by (8), sort(H) by  $d(pr_j)$

for  $R_j \in H$

    if (11) then (12),  $pr_j \leftarrow l^j$ , continue  $R_j$

    calculate  $s$  from (13)

    mov[ $R_k$ ]  $\leftarrow 0 \forall R_k \in H$

```

for  $R_k \in H$  in increasing deviation from  $R_j$  in  $H$ 
  calculate  $s_k$  from (14)
   $mov[R_k] \leftarrow \max(s, s_k), s \leftarrow s - s_k$ 
  if  $s \leq 0$ , break  $R_j$ 
end  $R_k$ 
if  $s > 0$ 
   $space\_per\_segment \leftarrow ((s_2 - s_1) - \sum_o wid(pr_o)) / (size(H) + 1)$ 
  for  $R_k \in H, l^k \leftarrow space\_per\_segment + [\sum_o(space\_per\_segment + wid(pr_o)) \forall R_o <$ 
   $R_k] + wid(pr_k)/2$ 
  break  $R_j$ 
else
  for  $R_k < R_j$  in  $H, pr'_k \leftarrow pr_k - \sum_{R_k < R_o < R_j} mov[pr_o]$ 
  for  $R_k > R_j$  in  $H, pr'_k \leftarrow pr_k + \sum_{R_j < R_o \leq R_k} mov[pr_o]$ 
   $pr'_k \leftarrow \text{left}(pr'_k) - sm - wid_k/2$ 
   $pr \leftarrow pr', l \leftarrow pr$ 
end if
end  $R_j$ 
return  $l$ 

```

### 4.2.3 Vehicle Trajectory Generation

In planning the trajectory for the expanded state the task is two-fold. First we need to make  $R_i$  move from the state  $l_i$  to the state  $l'_2$  which is obtained from state  $l^i$  as per the terminology followed in planning for a single vehicle. We then need to modify all vehicles in  $H - \{R_i\}$  to occupy their modified lanes, and to further ensure that no collision is recorded between any two vehicles.

We first move  $R_i$ . We select state  $l'_2$  at  $Z_2$ , with orientation perpendicular to  $Z_2$  and the same lateral position. The Clothoid curve is drawn from  $l_i$  to  $l'_2$  and a curve parallel to the road is drawn from  $l'_2$  to  $l^i$ . Both curves are checked such that all intermediate points in both curves belong to  $\zeta_{dynamic}^{free}(\tau^2)$ . Here  $\zeta_{dynamic}^{free}(\tau^2)$  denotes the free configuration space considering the dynamics of vehicles specified in plan  $\tau_2$  (equation (15)). At time  $t$ , the space is given by equation (16).

$$\tau^2 = \cup \tau_j \forall j, e_j < e_i, R_j \notin H \quad (15)$$

$$\zeta_{dynamic}^{free}(\tau_2, t) = \zeta_{static}^{free} - \bigcup_{j \in \tau^2} \tau_j^2(t) \otimes R_j \quad (16)$$

Local optimizations are done which include varying state  $l^i$  along  $Z$  in order to make the path feasible. On computing a feasible plan the plan vector  $\tau^2$  is updated to account for the computed trajectory of  $R_i$ .

The next task is to plan each of the vehicles excluding  $R_i$  in  $H$ . For any vehicle  $R_j$  we first update the lane information  $lane_j$ , as computed, and then attempt to make a trajectory by using the updated lane. The change is made by altering the lateral axis values of lanes. All lanes from  $lane_j[a+2]$  to  $lane_j[b]$  are given a lateral value of  $l^i$ . This lane is then used for the motion of the vehicle. Vehicles may be taken in any order in  $H$ . However once the trajectory of a vehicle is computed, it is added in the plan  $\tau_2$ . The complete algorithm is given by Algorithm 5. The expansion strategy is given in Algorithm 6. We simply count the number of lanes required and distribute the available space between the vehicles. This is followed by the generation of the necessary trajectories.

The approach is illustrated in Figure 8. Figure 11(a) meanwhile shows the discovery of another vehicle whose travel plan is shown in the same figure. Figure 11(b) shows the generation of the lanes. The planned trajectories are shown in Figure 11(c). It may be noted that both the lane and trajectory of  $R_i$  would not be as shown in Figure 8(b) and 8(c). This is because presently we assume that  $R_i$  stops after the planned path at  $l'_2$ . When we further expand the state of  $R_i, R_j$  this would give space for  $R_i$  to pass by, considering the extended motion as a result of subsequent expansions. In this manner  $R_i$  would keep moving with  $R_j$  to the side, as planning proceeds denoting motion of  $R_i$  along the road. This stops when  $R_i$  is sufficiently ahead of  $R_j$ .

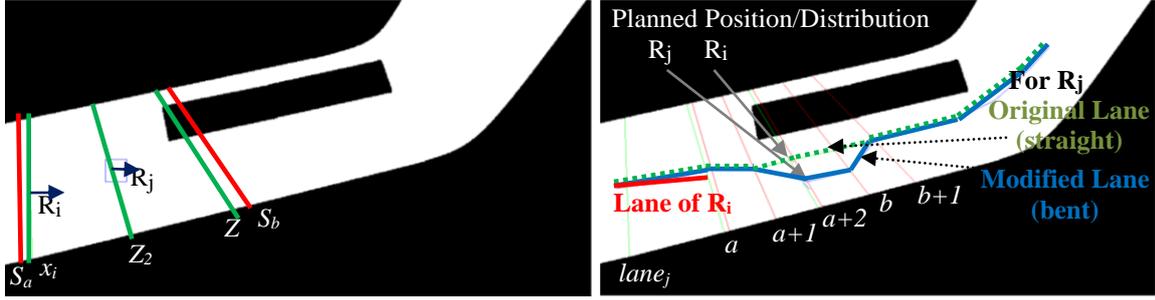


Figure 11(a): Computing lane vehicles      Figure 11(b): Distribution of lanes amongst vehicles (label colours correspond to line colours)

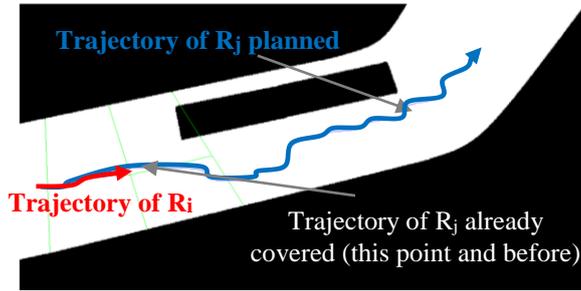


Figure 11(c): Generated trajectories for vehicles

**Algorithm 5: Trajectory generation from the current state to the expanded state**  
**GenerateTrajectory( $l, L_i$ )**

Calculate  $\tau^2$  using (15)

**Algorithm 5.1: GenerateTrajectorySelf()**

$[l^1, l^2] \leftarrow$  apply small deviations to  $l^1$  till  $connect_2(L_i, l^2) \in \zeta_{dynamic}^{free}(\tau_2) \wedge connect_2(l^2, l^1) \in \zeta_{dynamic}^{free}(\tau_2)$ , return NULL if still infeasible

$\tau_i^2 \leftarrow L_i[\tau_i] \cup connect_2(L_i, l^2)$

$l_2[lane_i] \leftarrow L_i[lane_i] \cup l^2$

$l_2[t] \leftarrow L_i[t] + \|connect_2(L_i, l^2)\|/v_i$

for  $R_j \in H$

$lane_j[l][Y] \leftarrow l^1 \forall a+2 \leq l \leq b$

**Algorithm 5.2: GenerateTrajectoryElse()**

$\tau_j^2 \leftarrow \tau_j(t) \forall t \leq lane_j[a+1][T]$

if  $\tau_j(t) \notin \zeta_{dynamic}^{free}(\tau^2, t) \forall L_i(t) \leq t \leq lane_j[a+1][T]$ , return NULL

```

for l ← a+2 to b

    lanej[l] ← apply small deviations to lanej[l] till connect2(lanej[l-1], lanej[l]) ∈  $\zeta_{dynamic}^{free}(\tau_2)$ ,
    return NULL if still infeasible

     $\tau_j^2 \leftarrow L_i[\tau_j] \cup \text{connect}_2(\text{lane}_j[l-1], \text{lane}_j[l])$ 

    lanej[l][T] ← lanej[l-1][T] + || connect2(lanej[l-1], lanej[l]) ||/vj

end l

```

end R<sub>j</sub>

l<sub>2</sub>[τ] ← τ<sub>2</sub>

return l<sub>2</sub>

### Algorithm 6 Free State Expansion Strategy

**FreeStateExpansion** (L<sub>i</sub>)

H ← getLaneVehicles(L<sub>i</sub>)

l ← getLanePositions(H, L<sub>i</sub>)

return GenerateTrajectory(l, L<sub>i</sub>)

### 4.3 Vehicle Following

The free state expansion attempts to make a vehicle move at its highest possible speed, which may not always result in an optimal plan. Consider that a vehicle  $R_i$  capable of high speeds finds a vehicle  $R_j$  moving slowly in front.  $R_i$  would naturally attempt to overtake  $R_j$ . But overtaking may not be possible due to various reasons like narrowing of road width, discovery of another vehicle, obstacle discovery, etc. In all these cases  $R_i$  has to slow down by significant amounts to avoid a collision. Slowing the vehicle iteratively to compute a possible trajectory is itself a time consuming activity.

The ideal travel plan in the above presented scenario would be planning  $R_i$  to *follow*  $R_j$  until the conditions become favourable for overtaking and to accommodate both vehicles in parallel for the required time. Keeping to the general algorithm methodology, we have a state  $L_i$  that we need to expand into state  $l_2$  using the vehicle following strategy. The implementation of this behaviour of the vehicle is simple. It involves two parts. Selecting the vehicle to follow and following the selected vehicle.

A vehicle aims to follow the vehicle going in the same direction that is closest to it. Separation is measured along the Y axis. It is however important that the selected vehicle  $R_j$  must be passing through  $P_a$ . Further the search for a potential  $R_j$  is always carried within a longitudinal distance  $2\Delta(v_i)$ . The other thing that we need to keep in mind in designing the vehicle following strategy is that vehicle  $R_i$  would ultimately be attempting to overtake  $R_j$ . In order for overtaking to be possible,  $R_i$  must have a sufficient longitudinal distance from  $R_j$ , which, as per our modelling, is  $\Delta(v_i^{max})$ . Vehicle following is done by reducing the speed of the vehicle  $R_i$  to a value of  $v_i^{red}$  given by equation (17).

$$v_i^{red} = \begin{cases} v_j & |\tau_j(t)[X] - L_i[X]| > \Delta(v_i^{max}) \\ v_j - \delta & \text{otherwise} \end{cases} \quad (17)$$

Here  $\delta$  is a small factor maintaining the difference in speeds. The value of speed  $v_j$  can be found by looking at  $R_j$ 's travel plan at the current time  $t$ .

The next task is making the vehicle move forward. We use exactly the same algorithm as in the free state expansion. The only difference is that the obstacle analysis line  $Z$  is constructed at a longitudinal distance of  $2\Delta(v_i)$ . However the vehicle placement line  $Z_2$  is constructed at a longitudinal distance of  $\Delta(v_i^{red})$ . Since the motion of  $R_i$  is with a speed of  $v_i^{red}$ , it is evident that all the steering needed would be possible. The task of

obstacle analysis at  $Z$  is to ensure that  $R_i$  avoids all static obstacles and other moving vehicles other than  $R_j$ . Since the width of  $R_i$  is different from  $R_j$ , the manner of avoiding vehicles and obstacles would be different for the two vehicles. While we compute the vehicles for the lane requirements,  $R_j$  having greater or equal speed to  $R_i$  would not qualify for an independent lane. The possibility of a vehicle following another vehicle was accounted for in the lane computation.

Consider the same situation as in Figure 8. Here  $R_i$  also needs to take the option of following  $R_j$ . As at the time of planning the distance between  $R_i$  and  $R_j$  is less than  $\Delta(v_i^{max})$ , the speed of  $R_j$  is taken to be slightly slower than the speed of  $R_i$ . In planning  $R_i$  does not consider  $R_j$  and travels as shown in Figure 12. Figure 12 also shows the travel plan of  $R_j$ . No collisions happen since  $R_i$  is travelling at a lower speed than  $R_j$ . The general algorithm is given in Algorithm 7.

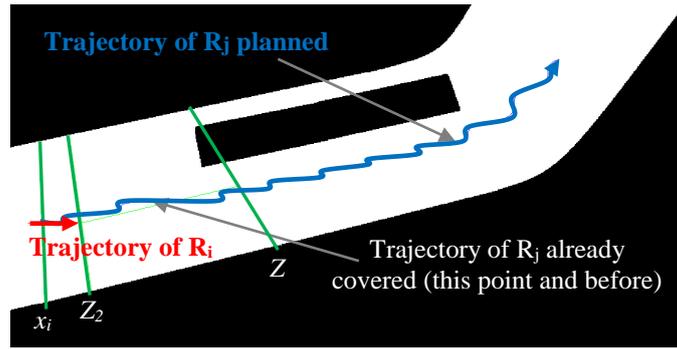


Figure 12: Trajectories of vehicles for vehicle following behaviour

#### Algorithm 7 Vehicle Following Expansion Strategy

##### VehicleFollowExpansion( $L_i$ )

Select closest  $R_j$  in  $Y$  axis, passing through  $P_a$  in the same direction

Calculate  $v_i^{red}$  from (17)

Return FreeStateExpansion( $L_i$ ) with  $v_i = v_i^{red}$  and  $Z$  at  $x_i + \Delta(v_i)$ ,  $Z_2$  at  $x_i + \Delta(v_i^{red})$

#### 4.4 Wait for Vehicle

The travel plan of multiple vehicles may not always involve all vehicles moving slowly or quickly in the road segment. It may in fact require a vehicle to *wait* for some time. As an example consider the situation of a narrow bridge. Since vehicles have variable width, the number of vehicles that may pass the bridge at same time without collisions is variable. Consider the case when a large vehicle  $R_j$  is already on the bridge, making it impossible for another vehicle  $R_i$  of decent width to fit if coming from the other side of the road. In such a case  $R_i$  needs to stop at a location before the bridge and wait for  $R_j$  to leave the bridge. The strategy is inspired by a general driving scenario in English traffic, wherein parked vehicles (obstacles) on small roads reduce the number of lanes from 2 to 1, requiring inbound and outbound vehicle drivers to coordinate between themselves deciding who goes first.

This stresses the need to make a vehicle wait at some point for some duration. It is computationally impossible to compute the time a vehicle must wait at every location by trying all possibilities in the search task. We make a simple heuristic that a vehicle must always wait for a vehicle to pass by. Using this heuristic the task of waiting may be broken down into steps: deciding the vehicle to wait for, adjusting the position for the other vehicle to smoothly pass by, modifying other vehicles' trajectory to account for the vehicle waiting, and computing the state after wait which becomes the expanded state as per this expansion strategy.

The general expansion algorithm expects the expansion strategy to expand the state  $L_i$  towards  $P_a$ . In this approach we do not place any restriction on the selection of the vehicle and all vehicles  $R_j$  that are tentative to pass through  $P_a$  at a later stage in their travel plan are worked for. Consider that  $R_i$  is waiting for  $R_j$  to pass through. Consider the set  $H$  containing all vehicles  $R_k$  (including  $R_j$ ) that pass through  $P_a$  from the other direction, earlier than  $R_j$ . The time when a vehicle  $R_k$  passes  $Z$  can be known by iterating through its travel plan.

##### 4.4.1 Calculating Preferred Positions

We know a stream of vehicles  $H$  would cross  $P_a$ , after which  $R_i$  may continue its journey. The next major issue is adjusting the position of  $R_i$ . Here we need to decide whether  $R_i$  would wait left or right of the stream of incoming vehicles. Similar situations in everyday life have vehicles waiting towards the left (countries having the left-side driving rule). However in certain cases of road and obstacles, there may be a need for vehicles to wait on the other side of the road, making the entire travel plan optimal. Assuming  $R_i$  would be able to maintain a distance of  $sm$ , for every  $R_k$  in  $H$ , we compute the ideal positions of  $R_i$  along the Y axis.

We assume the obstacle analysis line (here also called the *inside bridge line*)  $Z$  to lie in the region which was narrow and hence could not accommodate  $R_i$ . We further assume the vehicle placement line (here also called the *vehicle waiting line*)  $Z_2$  to be lying in the region which is wide enough for  $R_i$  to wait while the other vehicles in  $H$  pass by. Let  $l^k$  be the position of  $R_k$  when it crosses the line  $Z$  as per its plan. We assume in the modified plan of any  $R_k$  in  $H$ ,  $R_k$  would like to stay in the same lateral position till it crosses the waiting  $R_i$ . This means that any interesting behaviour of  $R_k$  would be suspended till it crosses  $R_i$  and it would need to travel parallel to the road while overcoming the waiting  $R_i$ . Hence while  $R_k$  is crossing the line  $Z_2$  at a point  $l^k_2$  it is to be found at the lateral position  $l^k[Y]$ . The preferred position of  $R_i$  considering only  $R_k$  is given by equation (18) and (19). The values denote the distance measured from the road boundary along  $Z_2$ . Here  $rl$  is the road width at  $Z_2$ .

$$\text{right}_k = l^k \cdot rl - \text{wid}_k/2 - sm - \text{wid}_i/2 \quad (18)$$

$$\text{left}_k = l^k \cdot rl + \text{wid}_k/2 + sm + \text{wid}_i/2 \quad (19)$$

Whether  $R_i$  attempts to wait on the left or right side of the stream, it needs to leave enough space for each vehicle in the stream to pass through, and hence the preferred points on both sides may be given by  $\min(\text{right}_k)$  and  $\max(\text{left}_k) \forall R_k \in H$ , out of which  $R_i$  chooses the one which requires least steering (equation (20)).

$$d(l^i_2) = \begin{cases} \min(\text{right}_k) & \text{abs}\left(\frac{\min(\text{right}_k)}{rl} - y\right) \leq \text{abs}\left(\frac{\max(\text{left}_k)}{rl} - y\right) \\ \max(\text{left}_k) & \text{abs}\left(\frac{\min(\text{right}_k)}{rl} - y\right) > \text{abs}\left(\frac{\max(\text{left}_k)}{rl} - y\right) \end{cases} \quad (20)$$

In case  $d(l^i_2)$  lies outside the road, it is made to lie inside with a separation of  $sm$ . Local optimization may be performed to induce feasibility, which gives the state to be expanded  $l_2$ . The state  $l_2$  however does not guarantee a distance of less than  $\Delta(v_i)$  from the obstacle ahead. It means further expansion of  $l_2$  may require reducing the vehicle's speed.

#### 4.4.2 Generating Trajectories

The next task is modifying the trajectories of each vehicle in  $H$  to account for the waiting  $R_i$ . Consider the segment of road along the X axis  $S = (x_i + \Delta(v_i) - \text{len}_i/2 - \Delta(v_k), x_i + \Delta(v_i) + \text{len}_i/2 + 2\Delta(v_k))$ . This is the region, at a distance of  $\Delta(v_k)$  on either side of  $l^i_2$ , which is affected by  $R_i$  waiting at  $l^i_2$  including the length of the waiting vehicle. In other words we consider  $R_i$  as an obstacle waiting at  $l^i_2$  and try to avoid  $R_k$  at a distance of  $\Delta(v_k)$ . We keep extra space to correct the position of  $R_k$  before surpassing  $R_i$  at  $l^i_2$  and further space after it has crossed waiting  $R_i$  to avoid collisions due to the immediate correction of position. Any lane of  $R_k$  that lies within this region is modified to account for the waiting  $R_i$ . As per the calculations of  $l^i_2$ , we expect  $R_k$  to have the same lateral position that it had when it crossed the line  $Z$ . Let  $R_k$  enter the region  $S$  at  $\text{lane}_j[a]$  and leave at  $\text{lane}_j[b]$ . The manner of changing the lane and then the trajectory is similar to free state expansion.

The order of planning of vehicles in  $H$  specifically takes planning of the vehicle being waited for  $R_j$  ahead of the other vehicles. This is because this planning gives us the time ( $tw$ ) when  $R_j$  crosses its  $\text{lane}_j[b]$  and hence  $R_i$  can continue its journey from this time onwards. Other vehicles in  $H$  consider  $R_i$  to be waiting only till the time  $tw$ . Hence any of their lanes are only modified if the vehicle crosses the position before this time.

The next task is generating a trajectory for  $R_i$  from  $L_i$  to  $l^i_2$ . The intention behind first generating trajectories of the vehicles in  $H$  and later  $R_i$  is that as we keep generating trajectories, they keep getting added to the travel plan  $\zeta_{dynamic}^{free}(\tau_2)$  which the vehicles being planned later must avoid. Local optimizations are performed on all vehicles in  $H$  and  $R_i$ . By this scheme we make clear that the path of vehicles in  $H$  must be as unchanged as possible, with  $R_i$  compromising as much as possible in terms of separation availability. Local optimizations in

trajectory generation of  $R_i$  ensures that any possibility of feasible curve generation of  $R_i$  returns a trajectory, in case there is no sufficient space. Computing the clothoid from  $L_i$  and  $l'_2$  and adding it into the expanded state of  $L_i$  is similar to free state expansion. The major difference is that the time a vehicle reaches  $l'_2$  is taken to be  $tw$  and not as computed by the general algorithm.

Figure 13(a) shows the stage when  $R_i$  discovers the sudden change in the width of the road, which initiates this expansion technique as per the heuristic (section 4.5). The region  $S$  for  $R_k$  is shown in the same figure. Figure 13(b) meanwhile shows the modification of the lane of  $R_k$ . Since a large amount of space was available and the vehicle was travelling in a straight line, no change was recorded. The trajectories of  $R_i$  and  $R_k$  are shown in Figure 13(c). The final waiting point for  $R_i$  becomes the initial state for further expansions. Note that in this figure the direction of motion of the vehicle is in the opposite direction as compared to the earlier figures. The general structure of the algorithm is similar to the free state expansion and is given as Algorithm 8.

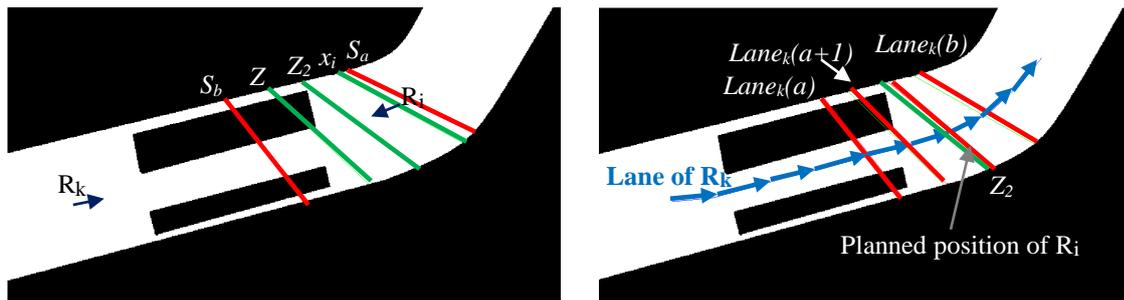


Figure 13(a): Detection of sudden narrowing of width of road      Figure 13(b): Modification of lane of  $R_k$

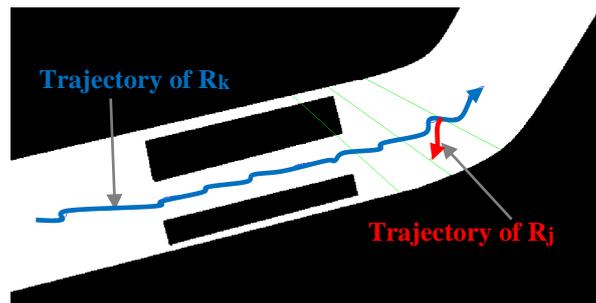


Figure 13(c): The generated trajectory for vehicles

#### Algorithm 8 Wait for Vehicle Expansion Strategy

##### WaitForVehicleExpansion( $L_i$ )

$l_2^s \leftarrow \text{NULL}$

for  $R_j : R_j$  crosses  $P_a$  in opposite direction to  $R_i$

    calculate  $H$

    calculate  $\tau^2$  using (15)

    calculate  $l'_2$  using (20)

    calculate  $lane_j[a]$  and  $lane_j[b]$

$lane_j[l][Y] \leftarrow l^k[Y] \quad \forall l \in (a, b)$

    success  $\leftarrow$  GenerateTrajectoryElse()

    if not success, continue  $R_j$

$tw \leftarrow lane_j[b][T]$

    for  $R_k \in H - \{R_i, R_j\}$

        calculate  $lane_k[a]$  and  $lane_k[b]$

$lane_k[l][Y] \leftarrow l^k[Y] \quad \forall l \in (a, b)$

        success  $\leftarrow$  GenerateTrajectoryElse()

        if not success, continue  $R_j$

    end  $R_k$

    success  $\leftarrow$  GenerateTrajectorySelf() from  $L_i$  to  $l'_2$  only

$l_2^s[T] \leftarrow tw$

    if not success, continue  $R_j$

```

    l2s ← l2s ∪ l2
end Rj
return l2s

```

## 4.5 Expansion Strategy

An easy method of implementation would be to use all three expansion strategies in all the states of the planning. However this results in a significantly large computational time. In order to reduce the complexity we decided to use heuristics to prune the expansions. Along with a natural driving analogy as well as general modelling of the planning scenario this helps us to mine out good scenarios. We use a heuristic rule common for the free state and vehicle following expansion, based on which either or both techniques may be used; and one for the wait for vehicle technique.

The decision whether to use the free state expansion strategy or a path following strategy is done on the basis of the vehicles in the critical region (say  $H$ ) that have been encountered in the expansion of state  $L_i$ . The vehicles that were encountered in a similar region when the state  $L_i$  was the expanded state are known, say  $L_i[H]$ . If the two sets  $L_i[H]$  and  $L_i[H]$  match, it means that planning is encountering the same vehicles again. However in case the two sets do not match, it means that a new vehicle has been encountered or  $R_i$  has completely passed some  $R_j$ . Now consider a road segment. By natural driving we know that once a vehicle  $R_i$  decides to overtake another vehicle  $R_j$ , it continues to do the same till overtaking is successful. Similarly if  $R_i$  decides to follow  $R_j$ , it continues to do so, till some vehicle  $R_k$  leaves, which was not giving space for  $R_i$  to overtake  $R_j$ . We make three observations here. First the decision whether to overtake a vehicle or to follow it is made on its first emergence. Second, once the decision has been made, it is unaltered as the vehicles go forward. Third, a change of decision is always marked by a vehicle entering or leaving. Vehicles entering and leaving in our algorithm are marked by a difference in the two sets. We know the strategy, out of the two strategies that resulted in the expansion of state  $L_i$ , say  $strategy(L_i)$ . A small assumption here is that every  $R_k$  is recorded at the critical region set. Alarming high differences in speed may make  $R_k$  unnoticed anytime in the critical region. This assumption can be improved by increasing the size of the critical region. The selected strategy of expansion is given in Algorithm 9.

### Algorithm 9

#### Expand( $L_i$ )

```

A ← NULL
for vi ← vmaxi to 1 in small steps
    calculate P using (2)
    for all Pa ∈ P
        calculate H, width(Pa)
        lookup H(Li), width(Li), strategy(Li)
        a, b, c ← NULL
        if H(Li) ≠ H
            a ← FreeStateExpansion(Li), strategy(a) ← FreeStateExpansion
            b ← VehicleFollowExpansion(Li), strategy(b) ← VehicleFollowExpansion
        else
            if strategy(Li) = FreeStateExpansion
                a ← FreeStateExpansion(Li), strategy(a) ← FreeStateExpansion
            if strategy(Li) = VehicleFollowExpansion
                b ← VehicleFollowExpansion(Li), strategy(b) ← VehicleFollowExpansion
        end if
        if width(Li) - width(Pa) > width_threshold
            c ← WaitForVehicleExpansion(L), strategy(c) ← WaitForVehicleExpansion
        A ← A ∪ a ∪ b ∪ c
    end Pa
    if A ≠ NULL, return A
end vi
return NULL

```

Similarly we place a heuristic rule for deciding whether the expansion by wait for vehicle technique needs to take place or not. Consider the obstacle analysis line  $Z$ . Let the length of obstacle-free line in  $P_a$  of state  $L_i$  be  $width(P_a)$ . Let  $width(L_i)$  be the length of the obstacle free line when  $L_i$  was produced. The technique of wait for vehicle expansion is only used if there is a significant (and sudden) decrease in the width. Hence a drop in values between  $width(L_i)$  and  $width(P_a)$  triggers this expansion technique. This is given in Algorithm 9.

## 5 Results

The complete system discussed was experimented with for a variety of scenarios. We first deal with understanding the vehicle dynamics in the presence of multiple vehicles. Our focus is on the three strategies presented, and hence the map taken does not have any obstacle. We then deal with understanding the vehicle dynamics within an obstacle network. For this part it is important to ensure that the vehicles do not collide or lie close to obstacles while displaying any behaviour.

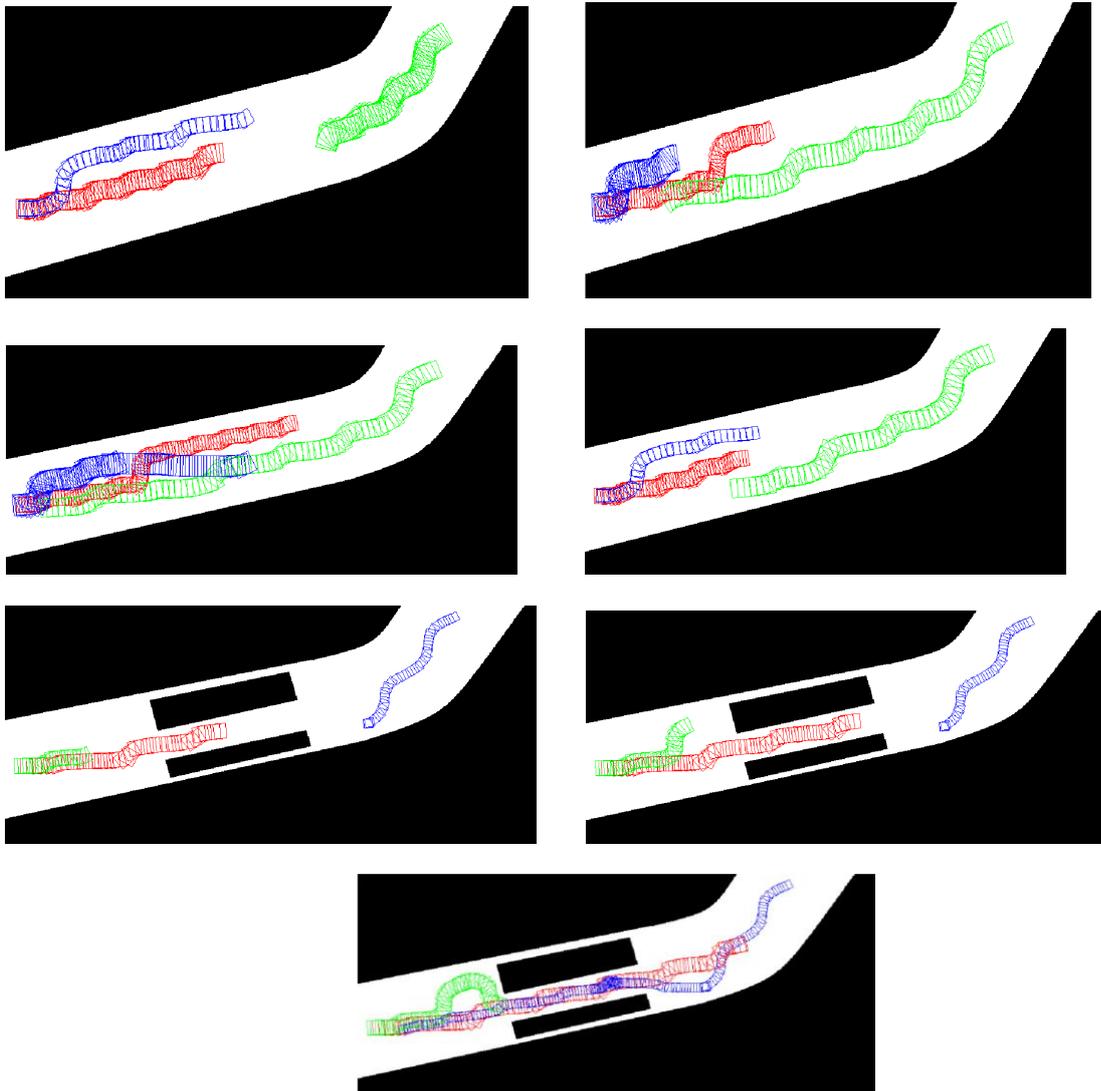


Figure 14: Experimental results with multiple vehicles without obstacles

### 5.1 Simple road experiments

We take a curved road with no obstacles. A number of vehicles enter the scenario from both ends at specified timings. The lengths and widths of all the vehicles were different. The first experiment involved studying the free state expansion strategy which leads to overtaking. To make the scenario complicated an additional vehicle

appears from the opposite end. The additional vehicle puts a strict restriction that the overtaking completes on time, else there is likely to be a collision. The instance of time when overtaking completes is shown in Figure 14(a). The trajectories denote the manner in which the vehicle emerges and goes forward with overtaking. It is clearly visible that the vehicle had to steer enough to avoid any collision and then move forward with its natural speed till it was sufficiently ahead of the vehicle.

The other behaviour was a vehicle following scenario which is also studied by similar experiments. Here again three vehicles are used. The oncoming vehicle allowed no possibility for the faster vehicle to overtake the slower vehicle. However, after the vehicle crossed, overtaking was possible. Figure 14(b) shows the situation where the oncoming vehicle crosses the two vehicles. Note that both vehicles drifted leftwards to ensure that no collision was recorded as well as a maximum separation being available. Soon after this vehicle crossed, the faster vehicle decided to overtake the slower vehicle. The intent to overtake is shown in Figure 14(c).

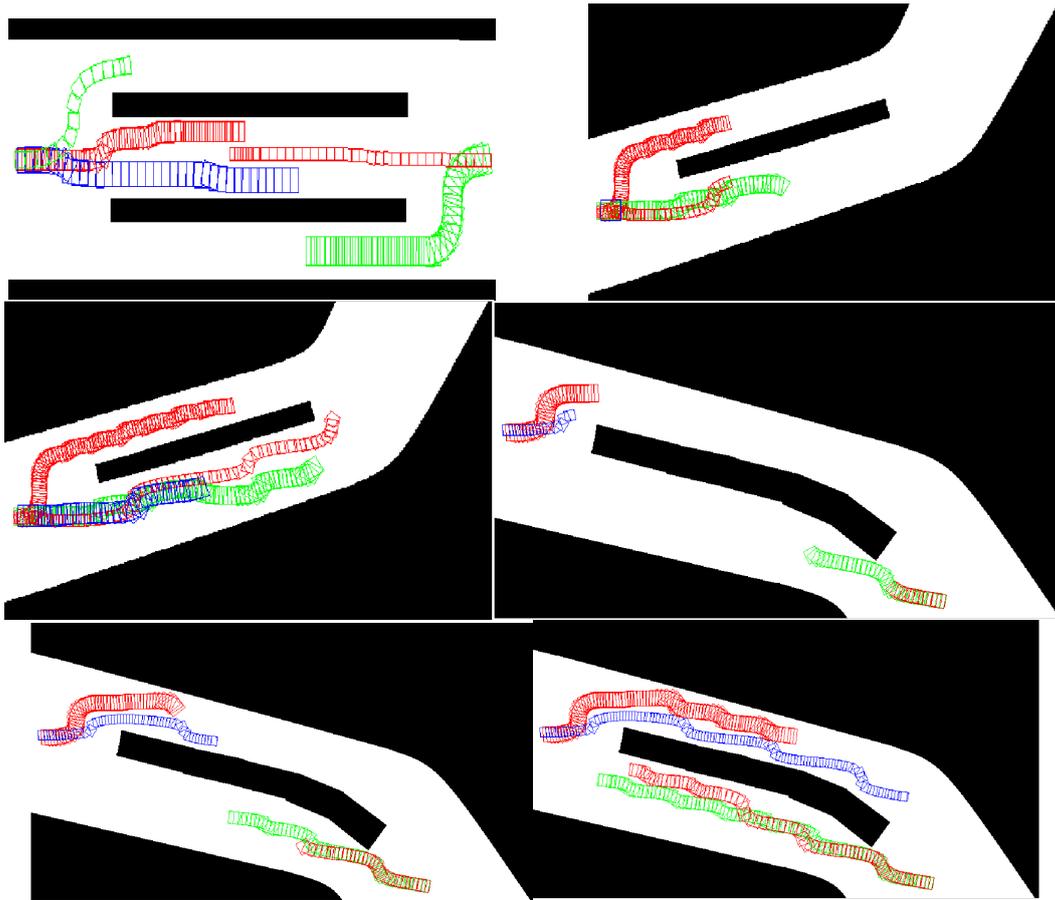
The different vehicles may need different lanes. Though we intent to align them in a way such that after steering and attaining their orientations they distribute the available road width amongst themselves, it is important to see whether the distribution takes place neatly. The next scenario presented three vehicles, two originating from one side of the road while the third from the other. The speeds were arranged and emergence timed so that the three met at some point in between. The result is shown in Figure 14(d). It can be seen that the separation between any two vehicles and road ends is kept large enough.

The last experiment was for the wait for vehicle strategy which requires build-up of a narrow bridge in the middle of the road. Here we chose three vehicles, two coming from one side and the other coming from the other side. The scenario and the results are shown in Figure 14(e) to 14(g). The first vehicle emerges and moves almost straight to cross the bridge. The second vehicle pre-computes that it needs to wait to avoid collision, it moves forward to take its waiting position as shown in Figure 14(e). Even before the first vehicle crosses, the third vehicle emerges into the planning scenario. It computes that it cannot cross the bridge before the first vehicle or just after the first vehicle, avoiding a collision with the second vehicle. So it also decides to wait for the second vehicle to cross the bridge by taking its position as shown in Figure 14(f). After the first vehicle has crossed, the second vehicle starts and similarly after the second vehicle has crossed, the third vehicle starts. This is shown in Figure 14(g).

## 5.2 Experiments of multi-vehicles in an obstacle network

The first scenario in this category was designed to look at the manner in which multiple vehicles select their paths. A broad straight road was divided into three lanes by physically placing long elongated obstacles. The vehicles were generated from either side of the road. The scenario and the solution are given in Figure 15(a). The first vehicle naturally selected the middle road ahead. As the second vehicle generated was wide, it could not travel straight in the middle road, and hence selected the bottom road. Now the other two vehicles generated preferred to stay in the middle road and occupy different lanes within this middle road, even though they could just fit in. At the time of their planning the top road was empty which was hence occupied by the next vehicle. Every vehicle was assigned a speed such that all the vehicles met or collided in the middle if travelling throughout by their speeds. This ensured no vehicle following behaviour was encountered.

In the experiments of section 5.1 we discussed vehicle behaviours in the absence of obstacles. However it may be essential for overtaking behaviour to complete within some obstacle sub-network. If a collision is encountered, the plan may alternatively choose vehicle following behaviour which is collision free. The scenario was set up with a barrier dividing a curved road. The vehicles were generated from the same side of the road. Scenario and results are shown in Figure 15(b) and 15(c). The first vehicle took the narrower segment and the second the broader one. The third vehicle had a higher speed and hence decided to overtake the slower vehicle in the broader segment. The intent to overtake is shown in Figure 15(b). Soon another vehicle was added in the scenario. Even though it had a high speed, it could not construct any overtaking strategy that completed within the segment with all the pre-planned behaviours of the other vehicles intact. So it decided to follow the front vehicle, while the earlier vehicle had almost overtaken. The situation is given in Figure 15(c).



**Figure 15: Experimental results with multiple vehicles in presence of obstacles**

The last scenario we take is to see if multiple behaviours can be simultaneously displayed. A curved road was taken with two vehicles of varying widths originated at different times from either side of the road. The scenario is displayed in Figure 15(d) to 15(f). On the first part of the road the vehicle generated later was of high speed and low size and was generated when the two vehicles were reasonably close by. The intent to overtake is shown in Figure 15(d). The vehicle is completely overtaken as shown in Figure 15(e). However on the other side of the road the two vehicles were separated by a large margin and the difference in speeds was kept low. As a result the second vehicle followed the first vehicle at high speed for a while as shown in Figure 15(e). Later it attempted to overtake it as displayed in Figure 15(f).

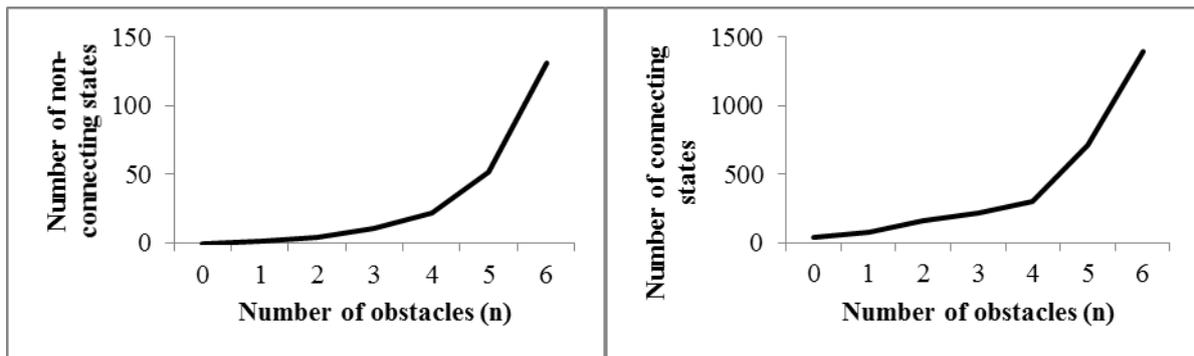
## 6 Discussion

Giving effective results as early as possible is one of the prime requirements of any planning algorithm. It is hence important to analyze the complexity of the algorithm, based on which the road segment length considered for planning may be decided. Too large a length may make the algorithm unable to give results within time, and too small a length would make planning ineffective, wherein different vehicle behaviours cannot be displayed and the vehicle can possibly be trapped in an obstacle network.

For  $n$  static obstacles on the road, going left or right of the obstacle may result in two different plans, indicating the total number of plans possible in the worst case to be  $2^n$  which the graph search explores. Consider a number of states produced by the graph search. A state may be called connecting state if it has one child or no children. These states do not multiply the search space or do not lead to new paths. Every obstacle approximately adds 1 non-connecting state corresponding to the state when it was first discovered in all possible paths. Hence for  $n$  obstacles, total paths in the worst case are  $2^n$  each approximately of length  $l$ , which makes the complexity  $O(2^n l)$ . The effect of factor  $\Delta(v_{max}^j)$  and connecting states on complexity is together communicated by the factor  $l$ . If some of these obstacles are small obstacles, there would be additional cost due to the reduction in speed of a vehicle which is done in steps.

The algorithm may also have multiple vehicles. Let  $a$  be the number of vehicles travelling in the same direction (inbound or outbound) as the vehicle being planned and  $b$  the number of vehicles travelling in the opposite direction. For a vehicles going in the same direction, it may either follow it or overtake it. Both possibilities need to be tried. The total possibilities to travel are  $2^a$  in the worst case. As per heuristics both possibilities were checked on encountering any vehicle. The possibilities are tried both when the vehicle arrives as well as when it departs. This may sometimes lead to a multiplicity of cases. The maximum number of possibilities in the worst case hence are  $2^a(a+2b)$ . In addition a vehicle may need to wait for another vehicle. Say the scenario has only one narrow bridge. At most the vehicle has to wait for all  $b$  vehicles coming, which means an additional  $b$  possibilities. However these possibilities are only tried at one particular state in the road when the width changes dramatically. Hence the addition of complexity is non-exponential. The complexity is hence  $O(2^a(a+2b))$ , which shows it is exponential for the number of vehicles in the same direction and obstacles. This number is usually very small considering the small road segment taken, or the vehicles would be simply following each other which is not accounted.

The resultant complexity becomes  $O(2^{n+a}(a+2b)l)$ , which means it is exponential in both  $n$  and  $a$ . We experimented both factors of obstacles and vehicles by a more realistic experimental scenario. First we took a straight road map and experimented on the effect of the addition of obstacles in the same. The intent was to expose the worst case that the algorithm may face, and hence all the obstacles were added one after the other. The road was significantly elongated so that we could fit in lots of obstacles. The effect of the increase in the number of connecting states  $s$  and non-connecting states  $n$  is shown in Figure 16. Since the vehicle was intentionally kept too small to allow avoiding all obstacles to be possible; many times after aligning to overcome an obstacle, in the next expansion stage it could further go to the other side of the obstacle, making possibilities per obstacles 3 rather than 2.

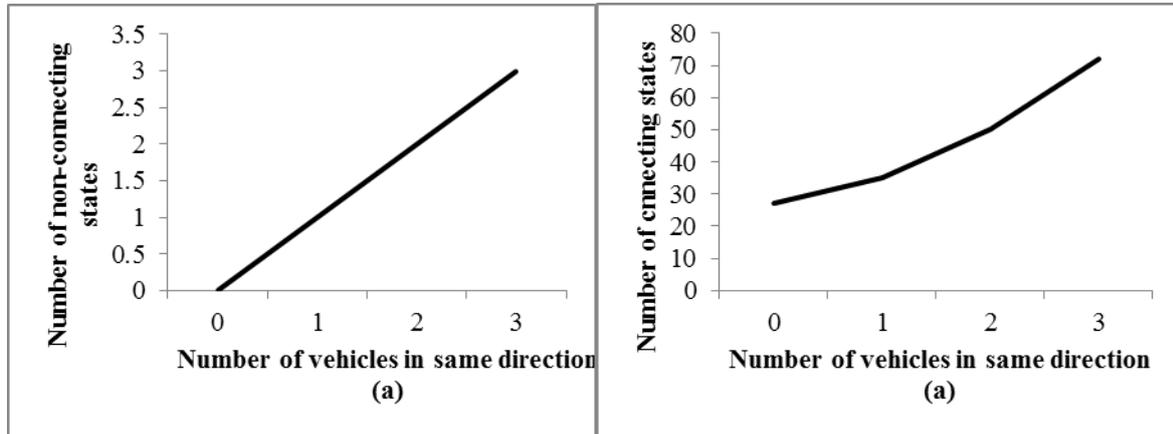


**Figure 16: Increase in number of (a) non-connecting states and (b) connecting states with increase in number of obstacles**

Similar experiment were performed with the number of vehicles. However here the worst case was found when the vehicle being planned, even if it decided to follow another vehicle, had possibilities of overtaking other vehicles while it was following. This is possible only when all  $a$  vehicles in front are arranged in increasing order of speeds with large gaps between them. The scenario was difficult to create even after prolonged increase in the road width. Clearly though theoretically possible, the situation is not practical. The practical worst case is to have a series of vehicles going at equal speeds separated by large gaps. For every vehicle, the vehicle being planned may decide to overtake, or to follow. If it overtakes, the choices repeat with the other vehicle in front. If it follows, no further choice is available. Hence the total number of cases possible are  $a+1$ , where  $a$  is the number of vehicles in front. Clearly the practical worst case is linear. The number of connecting and non-connecting states for an increasing number of vehicles is shown in Figure 17. The connecting states are not linear, as the section in which the vehicle being overtaken is recorded, and the road length remaining makes the difference.

The other factor is the longitudinal separation constant. This factor was kept as per the present notion that a vehicle must be able to steer significant distance in the Y axis. This puts a lower threshold on the value of the factor. Increasing this factor decreases steering and increases smoothness. However on increasing the factor, a small obstacle problem becomes harder to detect and rectify, close overtaking cannot be performed, and vehicles start aligning early on seeing an obstacle or some other vehicle. All these factors affect optimality. Too small a value however leads to too sharp turns and large computational time.

From the software point of view one of the greatest tasks ahead is to change the data structure in which paths and lanes are represented internally. The data structures must cater for fast query response. Heuristics can be added in the expansion for computational speedup. Post processing techniques may be added to optimize the trajectories as a whole and remove any oscillations visible in overtaking, lane changes, and while travelling on a curved road. Displaying vehicle behaviour at traffic merging, diversions and crossing scenarios is further to be done. The simulations can then be extended to an entire transportation system. The simulations ultimately need to be validated on real vehicles and real roads in order to prove their effectiveness. .



**Figure 17: Increase in number of (a) non-connecting states and (b) connecting states for increasing number of vehicles in the same direction**

## 7 Conclusions

Having large diversities in traffic, be it in terms of vehicle size, speed capabilities or road infrastructure, necessitates sophisticated planning techniques for the motion of vehicles. In this paper we considered the traffic scenario, and the planning of, multiple autonomous vehicles. The planning performed was based on a prioritized Uniform Cost Search, affecting re-planning of vehicles of higher priority which ensured cooperation. The algorithm could try all possibilities and come up with an optimal plan. Experiments over complex grids of obstacles and other vehicles showcased the effectiveness of the algorithm. Further vehicle behaviours of overtaking, vehicle following, and waiting for vehicle ensured optimal performance for a variety of scenarios.

A major question that arises out of the research is the validity of the present traffic rules. In an autonomous scenario, should the vehicles be allowed to collaboratively construct optimal plans which do not adhere to static traffic rules, which was the case in this approach? This leads to possibilities of more driving behaviours. The algorithm needs to ensure completeness and absence of deadlocks. Pedestrians and humans are desirable as obstacles, since they intelligently clear up with time, causing no deadlocks with autonomous vehicles. In the case of autonomous vehicles only, deadlocks need to be handled algorithmically. We have taken a positive step towards identifying all possible scenarios to ensure an absence of deadlocks, as depicted by the experimental results.

## Acknowledgement

The authors wish to thank the Commonwealth Scholarship Commission in the United Kingdom and the British Council for their support of the first named author through the Commonwealth Scholarship and Fellowship Program (2010) - UK through award number INCS-2010-161.

## References

[1] Bishop R (2000) A survey of intelligent vehicle applications worldwide. In: Proceedings of the IEEE Intelligent Vehicles Symposium, IEEE, pp 25-30.

- [2] Tsugawa S (2002) Inter-vehicle communications and their applications to intelligent vehicles: an overview. In: Proceedings of the IEEE Intelligent Vehicle Symposium, IEEE, pp 564-569.
- [3] Reichardt D, Miglietta M, Moretti L, Morsink P, Schulz W (2002) CarTALK 2000: safe and comfortable driving based upon inter-vehicle-communication. In: Proceedings of the IEEE Intelligent Vehicle Symposium, IEEE, pp 545-550.
- [4] Ma Y, Chowdhury M, Sadek A, Jaihani M (2009) Real-Time Highway Traffic Condition Assessment Framework Using Vehicle–Infrastructure Integration (VII) With Artificial Intelligence (AI). *IEEE Trans Intell Transp Syst* 10(4): 615-627.
- [5] Chang HJ, Kwak HJ, Park GT (2009) Efficient dissemination method for traffic jams information sharing based on inter-vehicle communication. In: Proceedings of the 2009 IEEE Student Conference on Research and Development, IEEE, pp 61-64.
- [6] Kala R, Warwick K (2012) Multi-Vehicle Planning using RRT-Connect. *Paladyn J Bahav Rob* 2(3): 134-144.
- [7] Rajamani R, Tan HS, Law BK, Zhang WB (2000) Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons. *IEEE Trans Control Syst Technol* 8(4): 695-708.
- [8] Varaiya P (1993) Smart cars on smart roads: problems of control. *IEEE Trans Automat Control* 38(2): 195-207.
- [9] Mohan D, Bawa PS (1985) An analysis of road traffic fatalities in Delhi, India. *Accid Annal Prev* 17(1): 33-45.
- [10] Paruchuri P, Pullalarevu AR, Karlapalem K (2002) Multi agent simulation of unorganized traffic. In: Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems: part 1, IEEE, New York, pp 176-183.
- [11] Vanajakshi L, Subramanian SC, Sivanandan R (2009) Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses. *IET Intell Transp Syst* 3(1): 1-9.
- [12] Furda A, Vlacic L (2011) Enabling Safe Autonomous Driving in Real-World City Traffic Using Multiple Criteria Decision Making. *IEEE Intell Transp Syst Magz* 3(1): 4-17.
- [13] Reveliotis SA, Roszkowska E (2011) Conflict Resolution in Free-Ranging Multivehicle Systems: A Resource Allocation Paradigm. *IEEE Trans Rob* 27(2): 283-296.
- [14] Schubert R (2012) Evaluating the Utility of Driving: Toward Automated Decision Making Under Uncertainty. *IEEE Trans Intell Transp Syst* 13(1): 354-364.
- [15] Schubert R, Schulze K, Wanielik G (2010) Situation Assessment for Automatic Lane-Change Maneuvers. *IEEE Trans Intell Transp Syst* 11(3): 607-616.
- [16] Kala R, Warwick K (2013) Planning Autonomous Vehicles in the Absence of Speed Lanes using an Elastic Strip. *IEEE Trans Intell Transp Syst*. doi: 10.1109/TITS.2013.2266355.
- [17] Kuwata Y, Karaman S, Teo J, Frazzoli E, How JP, Fiore G (2009) Real-Time Motion Planning With Applications to Autonomous Urban Driving. *IEEE Trans Control Syst Technol* 17(5): 1105-1118.
- [18] Chu K, Lee M, Sunwoo M (2012) Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles. *IEEE Trans Intell Transp Syst* 13(4): 1599-1616.
- [19] Gehrig SK, Stein FJ (2007) Collision Avoidance for Vehicle-Following Systems. *IEEE Trans Intell Transp Syst* 8(2): 233-244.
- [20] Sánchez-Ante G, Latombe JC (2002) Using a PRM Planner to Compare Centralized and Decoupled Planning for Multi-Robot Systems. In: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, Washington, DC, pp 2112 – 2119.
- [21] Lu Y, Huo X, Arslan O, Tsiotras P (2011) Incremental Multi-Scale Search Algorithm for Dynamic Path Planning With Low Worst-Case Complexity. *IEEE Trans Syst Man Cybern Part B: Cybern* 41(6): 1556-1570.
- [22] Arai T, Ota J (1992) Motion Planning of multiple mobile robots. In: Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Raleigh, North Carolina, pp 1761-1768.
- [23] Svestka P, Overmars MH (1995) Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In: Proceedings of the 1995 IEEE International Conference on Robotics and Automation, IEEE, Nagoya, Japan, pp 1631-1636.
- [24] Bennewitz M, Burgard W, Thrun S (2002) Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Rob Auton Syst* 41(2-3): 89–99.
- [25] Kala R, Shukla A, Tiwari R (2011) Robotic path planning in static environment using hierarchical multi-neuron heuristic search and probability based fitness. *Neurocomputing* 74(14-15): 2314-2335.
- [26] Lumelsky VJ, Harinarayan KR (1997) Decentralized Motion Planning for Multiple Mobile Robots: The Cocktail Party Model. *Auton Rob* 4(1): 121-135.
- [27] Chand P, Carnegie DA (2012) A two-tiered global path planning strategy for limited memory mobile robots. *Rob Auton Syst* 60(3): 309-321.
- [28] Cowlagi RV, Tsiotras P (2012) Hierarchical Motion Planning With Dynamical Feasibility Guarantees for Mobile Robotic Vehicles. *IEEE Trans Rob* 28(2): 379-395.

- [29] Kala R, Warwick K (2013) Multi-Level Planning for Semi-Autonomous Vehicles in Traffic Scenarios based on Separation Maximization. *J Intell Rob Syst*. DOI:10.1007/s10846-013-9817-7.
- [30] Leroy S, Laumond JP, Siméon T (1999) Multiple path coordination for mobile robots: A geometric algorithm. In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence, IEEE, Stockholm, Sweden*, pp 1118–1123.
- [31] Bayazit OB, Lien JM, Amato NM (2002) Probabilistic roadmap motion planning for deformable objects. In: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, IEEE*, pp 2126- 2133.
- [32] Nilsson NJ (1971) *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill, New York.
- [33] Russell SJ, Norvig P (2010) *Artificial intelligence: a modern approach*, Prentice Hall, New Jersey.
- [34] Alvarez-Sanchez JR, de la Paz Lopez F, Troncoso JMC, de Santos Sierra D (2010) Reactive navigation in real environments using partial center of area method. *Rob Auton Syst* 58(12): 1231-1237.
- [35] Nutbourne AW, McLellan PM, Kensit RML (1972) Curvature profiles for plane curves. *Comput Aided Des* 4(4): 176–184.
- [36] McCrae J, Singh K (2009) Sketching piecewise clothoid curves. *Comput Graph* 33(4): 452-461.
- [37] Peng J, Akella S (2003) Coordinating the Motions of Multiple Robots with Kinodynamic Constraints. In: *Proceedings of the 2003 IEEE International Conference on Robotics & Automation, IEEE, Taipei, Taiwan*, pp 4066-4073.
- [38] Peng J, Akella S (2005) Coordinating Multiple Robots with Kinodynamic Constraints Along Specified Paths. *Int J Rob Res* 24(4): 295-310.