A Scalable MDP-based Sensing and Processing Framework for Vehicular Networks

Rajarshi Chattopadhyay and Chen-Khong Tham
Department of Electrical and Computer Engineering
National University of Singapore
E-mail: e0009104@u.nus.edu, eletck@nus.edu.sg

***Abstract*—Smart Vehicles (SV) generally have on-board processing, wireless communication and sensing capabilities. They are useful for vehicular applications like autonomous driving (AD) or in-vehicle augmented reality (AR). One way to enhance the performance of these applications is by enabling SVs to exchange messages and sensor readings with other nearby SVs. However, due to the mobility and resource limitation of SVs, the availability of sensing and processing resources of nearby SVs and the communication links between SVs (V2V) are unreliable. Thus, it is a challenge to develop efficient sensing and processing schemes for SVs. We first propose a Markov decision processes (MDP) based sensing and proposing framework for SVs. The framework models the uncertainties and can be solved to obtain an optimum sensing and processing policy. Second, we point out that the spatial information required to obtain sensing policy, needs an extremely big state space for representation, making this framework unscalable. Thirdly, we propose separating the sensor selection problem from the sensing and processing scheme and explain how this simplifies the problem and makes it scalable. We then propose a maximum flow minimum cost based sensor selection heuristic. Finally, we compare the performance obtained by applying our heuristic with that of our original MDP based scheme. The results show that our heuristics performance is nearly as good as the original scheme while also increasing scalability.**

***Index Terms*—Vehicular ad-hoc Networks; Autonomous Driving; Distributed sensing; Edge Computing; Markov decision process; Policy iteration**

I. Introduction

Vehicles today are increasingly being equipped with an array of sensors, processing and communication capable devices and have constant access to the Internet. This has resulted in a new era of Smart Vehicles (SV). SVs can exchange messages with other nearby SVs and sense the environment using its sensors. This gives them the ability to cooperate and make intelligent decisions for preventing accidents, congestion and pollution.

Autonomous driving (AD) and in-vehicle augmented reality (AR) information systems are two SV technologies that have made a lot of progress in the past decade. The objective in AR systems is to use technology to enhance the senses of the driver. Traffic information may be obtained from the internet or from the SVs own sensors. Once obtained, the information is usually processed into a form more perceptible to the driver and displayed appropriately. The goal of AD is to make vehicles capable of functioning without a driver. Cameras and light detection and ranging (LiDAR) sensors are the most commonly used sensors in SVs for these applications [1], [2]. Typically, the raw sensor data is passed to a classifier for classification and motion detection. State of the art machine learning models are computationally expensive. Several works in literature have contributed towards developing small sized accurate and energy efficient models [3], [4] for embedded systems. The cloud-enabled vehicular networking paradigm [5] incurs high latency.

In the case of in-vehicle AR the nearby objects can be displayed to driver along with their motion and semantic information, who then decides the vehicle's trajectory. On the other hand, in AD trajectory planning is also done computationally. A typical trajectory planner [6] follows two main steps, global trajectory planning and local trajectory planning. The global trajectory is computed based on the initial location, the final destination, and the map of the region. Local trajectory planner computes slight deviations along the global trajectory. Typically, the local trajectory planning algorithm considers only the nearby objects within a predefined region of interest (RoI). Naturally, the trajectory planner could make a more informed decision if we increase the RoI. Sometimes the field of view (FoV) of the sensors may be limited by the presence of obstacles. This scenario is quite common in case of vehicles, e.g. the FoV of the vehicles front camera being blocked by the bus in front of the vehicle. However, we note that a vehicle maybe able to sense a much larger RoI with the help of the sensors of other nearby vehicles. V2V communication technology can help in communicating sensor data between vehicles.

This paper makes the following contribution. To the best of our knowledge, our work is the first to propose a MDP based framework that jointly considers sensing and processing in vehicular networks (VN). Secondly, we propose a maximum flow minimum cost based heuristic for making the framework scalable. The rest of the paper is organized as follows. Section II describes the system formulation. Section III presents our MDP based framework and Section IV presents the proposed heuristic. Section V states the simulation setting and results. Section VI describes the related works and Section VII concludes the paper.

II. System and Problem Formulation

A. Considered Scenario

Figure 1 depicts the scenario. Node U represents the user running the application. The largest possible square represents the RoI of the user. Let this region be denoted by R. The

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

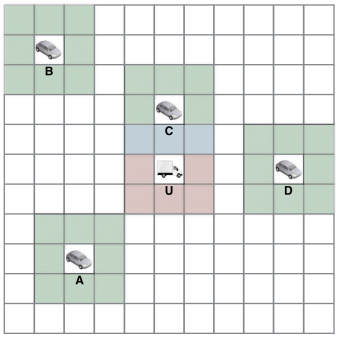


Fig. 1. RoI of user and FoV of available sensors

performance of node U's application depends on the quality and coverage of sensing within R. A, B, C and D are some other SVs (other nodes) within the wireless communication range of U. The RoI is subdivided into smaller sub-regions. Let the sub-regions be denoted by r_k. In this work we assume that the field of view (FoV) of a single sensor is one sub-region. The shaded region around a node denotes the combined FoV of all its sensors. The regions shaded red and green depict the FoV of node U and the other nodes respectively. The blue region shows an area that is within the FoV of senors of both nodes U and C. Clearly, node U can sense a greater portion of the RoI with the help of other nodes.

B. Sensing model

Sensor data utility depends on both the location of the sensor and its data quality [7]. Here location refers to the FoV. Quality may be measured based on the information content or some other attribute e.g. frame rate or resolution of video from SV cameras. We abstract sensor data quality of sensor j of node i as a discrete variable $q_j^i \in \{0,1,2,...,q_{max}\}$. $q_j^i = 0$ means that sensor j of node i is not sensing. The variable $\eta_j^i \in \{0,1\}$ represents the processing decision. $\eta_j^i = 1$ denotes that raw data is sent from sensor j of SV i to the user SV, whereas $\eta_j^i = 0$ denotes that the data from sensor j of SV i will be processed at SV i and only the result sent to the user SV.

C. Processing model

Semantic and motion segmentation of images involve linear operations on the image pixels. We define a unit task as the amount of computation required to process a sensor sample whose $q_j^i = 1$. Hence, the amount of computation required to process a sensor sample with $q_j^i = k$ can be considered equivalent to k dependent tasks. The processing model of SVs have two components, the task queue and the processor. A SV may receive raw sensor samples either from its own sensors or from the sensors of neighbouring SVs. Upon arrival the raw sensor samples are placed in the task queue. Assume that the task queue can not store more than M tasks. We use m^i to represent the number of tasks at the beginning of a time slot. If the q_j^i of an arriving sensor sample is greater than $M - m^i$, then it is dropped. The on-board processor on SVs is resource constrained and shared by several applications with unpredictable demands. We model the processor's capacity as $p^i \sim Poisson(\lambda^i)$ to account for the unpredictable demands. For simplicity, we assume $\lambda^i = \lambda \ \forall i$.

D. Communication Model

A V2V link between SVs is a wireless communication link and is affected by path loss, shadowing, fading and node mobility. Let P_{tr} denote the transmission power and P_{th} the receiver power threshold.

1) Probability of Link Existence: Let P_{rec} denote the received signal power at the receiver. As a consequence of shadowing and fading, P_{rec} at any given distance from the transmitter is a random process. The quantity of interest is $P(P_{rec} \geq P_{th}|x)$. It denotes the probability that $P_{rec} \geq P_{th}$ at a distance x from the transmitter, i.e. it is the probability of link existence at a distance x from the transmitter.

2) Effective transmission range: $P(P_{rec} > P_{th}|x)$ is monotonically decreasing w.r.t. respect to x. We define the effective transmission range of a transmitter, denoted by R_e, to be the distance from transmitter beyond which $P(P_{rec} > P_{th}|x) < 0.1$. Beyond R_e we assume that $P(P_{rec} > P_{th}|x) = 0$ [8].

E. Mobility model

1) Location state of neighbouring SVs: The location state of SV i, denoted by l^i, is the sub-region where SV i is currently located. Each sub-region is a square with side length S_L. Let x_i and y_i be the relative distance of SV i from user SV in the X and Y direction (two perpendicular directions) respectively. Then $l^i = b_{jk}$, only if $j = \text{ceil}(\frac{x_i}{S_L})$ and $k = \text{ceil}(\frac{y_i}{S_L})$. Here both $-R_e \leq x_i \ , \ y_i \leq R_e$.

2) Location State Transition Probabilities: Due to relative motion of the SVs, the location state of each neighbouring SV changes after each time interval. $P_{b_{ij}}^{b_{rs}}$ denotes the probability of the location state transition from state b_{ij} to state b_{rs} in one time interval. We do not describe a detailed mobility model here due to space constraint. We have briefly mentioned it in Section V.

3) Probability of Link Existence for a location state: $P(P_{rec} > P_{th}|x)$ at any point depends on the relative distance x from the transmitter. For a given location state b_{jk}, $P(P_{rec} > P_{th}|x)$ must be averaged over the concerned region as done in (1).

$$P_{LE}^{jk} = \int_{(j-1).S_L}^{j.S_L} \int_{(k-1).S_L}^{k.S_L} \frac{1}{S_L^2}.P(P_{rec} > P_{th}|\sqrt{x^2+y^2}) \ dxdy \quad (1)$$

III. MDP based framework for Sensing and Processing

A. State Space

The state space is defined as in (2). Here $M = (m^1,...,m^N)$ and $L = (l^1,...,l^N)$ are vectors of the queue states and the location states of all the neighbouring SVs respectively. m^U denotes the queue state of the user node.

$$\mathbb{S} = \{s = (M, L, m^U)\} \quad (2)$$

B. Action Space

The action space is defined as in (3). $Q = \{q_j^i | \forall \text{ i,j}\}$ and $\eta = \{\eta_j^i | \forall \text{ i,j}\}$ are vectors of the decision variables. Let $\mathbb{A}(s) \subseteq \mathbb{A}$ denote the set of actions possible from state s. An action is possible from a state if none of the SV queues overflow. These conditions are stated concisely in (4).

$$\mathbb{A} = \{a = (Q, \eta)\} \tag{3}$$

$$\begin{aligned} &(m^U + \sum_{j \in Se(U)} q_j^U + \sum_{k=1}^{N} \sum_{l \in Se(k)} \eta_l^k q_l^k) \leq M \\ &\left[m^i + \sum_{j \in Se(i)} (1 - \eta_j^i) q_j^i\right] \leq M;\ \forall i \in \{1, 2, ..., N\} \end{aligned} \tag{4}$$

C. State Transition Probabilities

Here we calculate the probability that the system transits from a state $s = (M, L, m^U)$ to a state $s' = (M', L', m'^U)$ after taking an action $a = (Q, \eta)$. Given s, s' and a the number of tasks processed by the user node U is $x^U = (m^U + \sum_{j \in Se(U)} q_j^U + \sum_{k=1}^{N} \sum_{l \in Se(k)} \eta_l^k q_l^k - m'^U)$ and by any other neighbouring node i is $x^i = (m^i + \sum_{j \in Se(i)} (1 - \eta_j^i) q_j^i - m'^i)$. Since $p^i \sim \text{Poisson}(\lambda)$, the probability that x^j tasks are processed by a node j during one period is

$$f(x^j, \lambda) = (\lambda)^{x^j} . \frac{e^{-\lambda}}{x^j!},\ \ x^j > 0 \tag{5}$$

Due to the independence of the number of tasks processed by each node we have:

$$P(M', m'^u | s, a) = f(x^U, \lambda^U) . \prod_{j \in \{1,2,...,N\}} f(x^j, \lambda^j) \tag{6}$$

Furthermore, the mobility of a node is independent of the mobility of other nodes. Therefore, we can write the probability of transition of L to L' as:

$$P(L' | s, a) = \prod_{j \in \{1,2,...,N\}} P_{l^i}^{l'_i} \tag{7}$$

Therefore, the overall transition probability from state s to s' given action $a \in \mathbb{A}(s)$ can be written as follows

$$P(s' | s, a) = P(M', m'^u | s, a) . P(L' | s, a) \tag{8}$$

D. Reward function

The overall reward given a state s and action $a \in \mathbb{A}(s)$, denoted by $R(s,a)$, is composed of the application utility and the costs involved.

1) Application Utility: Our application utility function, denoted by $U(s,a)$, is as given in (9). Here $u(q_j^i, l^i)$ denotes the data utility drawn from sensor j of node i after being processed. We assume that the utility drawn from a sensor sample is inversely proportional to the time spent since its arrival. $f_{De}^U(s,a)$ and $f_{De}(s,a,i)$ in (9) account for this. No utility is drawn if the sensor data is lost during communication. The first term within the double summation in the expression for U(s,a) in (9), calculates the utility when $\eta_j^i = 1$. C and $I(\ )$ denote the channel capacity and the identity function respectively. It is clear that ψ_j^i in (9) will be positive only when the size of raw sensor data from sensor j of node i (X_j^i) is less than the effective channel capacity. The second term within the double summation in the expression for U(s,a) in (9) calculates the utility when $\eta_j^i = 0$, i.e. the processed result is sent to user over cellular network.

$$\begin{aligned} U(s,a) &= \alpha . \sum_{i=1}^{N} \sum_{j \in S(i)} u(q_j^i, l^i) \Big[\eta_j^i . I\left(\psi_j^i\right) . f_{De}^U(s,a) \\ &\quad + (1 - \eta_j^i) . f_{De}(s,a,i)\Big] \\ \psi_j^i &= C . \Delta t . P_{LE}^{l^i} - X_j^i \\ f_{De}^U(s,a) &= \frac{\lambda^U}{1 + m^U + \frac{1}{2} . \left[\sum_{i=1}^{N} \sum_{j \in Se(U)} \eta_j^i q_j^i + \sum_{j \in Se(U)} q_j^i\right]} \\ f_{De}(s,a,i) &= \frac{\lambda^i}{1 + m^i + \sum_{j \in Se(i)} (1 - \eta_j^i) q_j^i} \end{aligned} \tag{9}$$

2) Cost Functions: The user incurs cost due to energy consumption, incentives given to user etc. Let $C(s,a)$ denote this cost. We do not choose a particular $C(s,a)$ at the moment. We define another function $Pl(s,a)$. We let $Pl(s,a) = -\infty$ if two sensors with the same FoVs have non-zero q_j^i's. In all other cases $Pl(s,a) = 0$. Now putting both the parts together we define the reward function $R(s,a)$ in Equation (10).

$$R(s,a) = U(s,a) + Pl(s,a) - C(s,a) \tag{10}$$

After making suitable choices for the unspecified functions, the above MDP based framework can be solved using either value or policy iteration for obtaining an optimum sensing and processing policy for SVs.

IV. Maximum Flow Minimum Cost based Sensor Selection Approach

The MDP based framework jointly selects the optimum q_j^i and η_j^i. However, the dimensionality of the state space is very high. Let N_{sr} denote the total number of sub-regions within the RoI. Then the dimensionality of the state space is $M^{N+1} . N_{sr}^N$. Clearly, the framework does not scale well with N_{sr}. In this section we propose a heuristic with a goal of avoiding these problems.

l^i is only required for computing $u(q_j^i, l^i)$ and $Pl(s,a)$. P_{LE}^{jk} depends is on $\sqrt{x^2 + y^2}$, i.e. relative distance, rather than on (x,y). To make our state space independent of L we make two adjustments. Firstly, we assume that $u(q_j^i, l^i)$ is of the form $u(q_j^i, l^i) = u_q(q_j^i) . u_l(l^i)$. Secondly, we remove $Pl(s,a)$ from our framework, but ensure that no two sensors with $q_j^i \neq 0$ have the same FoV.

1) Probability of Link Existence for relative distance states: Just like location states, we can partition the relative distance between two SVs into K quantized states of width $\omega = \frac{R_e}{K}$. The

average probability of link existence for a distance state is as given in (11). Using this our framework will be independent of L.

$$P_{LE}^{i} = \int_{(i-1).\omega}^{i.\omega} \frac{1}{\omega}.P(P_{rec} > P_{th}|x)\ dx \tag{11}$$

2) Sensing Task Definition: Here we define the problem of sensing the sub-regions within the RoI as a collection of tasks. We will refer to these tasks as T_S. Each T_S involves sensing one sub-region. Thus, we can define them as in (12). Let N_{Ta} denote the total number of sub-regions.

$$T_S^x = \{r_l\}\ \forall x \text{ st. } r_x \in R \tag{12}$$

3) Cost of Sensing Tasks: $u_l(l^i)$ denotes the location based value of a sensor whose location information is in l^i. We define the location cost of task T_S^x to be $\frac{1}{u_l(T_S^x)}$.

4) Worker Nodes Definition: The SVs within the wireless communication range of the user can be considered as potential worker nodes. Equation (13) defines each SV within the wireless communication range of the user as a worker node w_i. Here $R_i = \{r_k,..,r_s\}$ is a vector of FoVs of node i sensors. Let T_{max} denote the maximum number of sensors that can be hired from any single neighbour. Nodes may use such limits to control energy consumption.

$$w_i = \{R_i, T_{max}\} \tag{13}$$

5) Allocating Sensing Tasks to Worker Nodes: In terms of worker and task definitions in (13) and (12), selecting the sensors for each $S(i)$ is equivalent to assigning those sensors the task of sensing their own FoVs. It is just like mapping a set of tasks to worker resources, the worker being the neighbouring SVs and their sensors being the worker resources. We must also ensure that no two selected sensors have the same FoV.

6) Maximum Number of Selected Sensors: Typically, the user node does not prefer paying more than a certain amount of incentive per time step. Let κ represents the sensing cost per time slot when $q_j^i = 1$. Ignoring the processing and communication costs, the minimum cost to use the selected sensors will be $\kappa.\sum_{i=1}^{N}|S(i)|$. This justifies the need for a maximum limit on the number of selected sensors.

7) Reduction to the Minimum Cost Maximum Flow Problem:

Definition 1. *(Maximum Task Assignment (MTA)). Let $|A_n|$ be the number of tasks that the user wants to process during time slot t_n. The MTA is the process of assigning tasks to the workers during t_n, such that the number of assigned tasks is maximized, while still being less than or equal to $|A_n|$.*

Definition 2. *(Minimum Cost Maximum Task Assignment (MCMTA)). The MCMTA is the process of assigning tasks to worker nodes during one time slot, such that the number of tasks assigned is equal to the number of tasks assigned by MTA, while the location cost of the assigned tasks i.e. $\sum_{i=1}^{N}\sum_{j\in S(i)}\frac{1}{u_l(l^{ij})}$ is minimized.*

We know from Definition 2 that that the solution of MCMTA depends on the solution of MTA. It turns out that MTA and MCMTA are jointly reducible to the minimum cost maximum flow problem. We prove this Theorem 1.

Theorem 1. *MTA and MCMTA are jointly reducible to the minimum cost maximum flow problem.*

Proof. At any time slot let $W = \{w_1, w_2, ..\}$ be the set of workers and $T = \{T_S^1, T_S^2, ..., T_S^{N_{Ta}}\}$ be the set of tasks. Let $G = (V,E)$ be the flow network graph with V as the set of vertices, and E as the set of edges. The set V contains $(|W| + N_{Ta} + 3)$ vertices. Each worker w_j maps to a vertex v_{1+j}. Each task t_j maps to a vertex $v_{1+|W|+j}$. We create a new source vertex labeled as v_0, another new vertex labelled v_1 and a new destination vertex labeled as $v_{|W|+N_{Ta}+2}$.

The set E contains $|W| + \sum_{i=1}^{|W|}|R_i| + |T| + 1$ edges. Recall that R_i is the vector of the FoV of worker i sensors. There are $|W|$ edges connecting v_1 to $v_{1+j}\forall j \in \{1,2,..,|W|\}$. For a given edge connecting v_1 to $v_{1+j}\forall j \in \{1,2,..,|W|\}$, we set the capacity to T_{max} since a maximum of T_{max} sensors can be hired from one worker node. There are also N_{Ta} edges connecting $v_{1+|W|+j}\forall j \in \{1,2,...,N_{Ta}\}$ to $v_{|W|+N_{Ta}+2}$. We set the capacity of each of these edges to 1, since each task has to be assigned to at most one worker. Further, there are $|R_j|$ edges from each vertex v_{1+j} to the corresponding vertices $v_{1+|W|+i}\forall i\ s.t.\ T_S^i \in R_j$. For each of these edges we set the cost to $d_{T_S^i}$ and the capacity to 1. Recall that T_S^i represents the sensing sub-region of sensing task i and d_{r_i} denotes the distance between the user node and the center of sub-region r_i. Lastly, we set the capacity of the edge between vertices v_0 and v_1 to $|A_n|$, the number of tasks that the user wants to process during the concerned time slot. All edges with unspecified cost have a default cost value of 0. It is easy to see that solving the maximum flow minimum cost problem for this flow network graph is equivalent to assigning tasks according to MTA and MCMTA. □

Figures 2a and 2b show an example scenario and its reduction to a flow network graph respectively. The RoI consists of 9 sub-regions, each labelled with a number in the lower left hand corner. The big square represents the user node. The big upper and lower triangles represent two worker nodes. The small upper and lower triangles in the upper-right and lower-right corners of certain sub-regions represent the fact that the respective worker node has this sub-region as the FoV of one of its sensors. Let $c(v_i,v_j)$ and $co(v_i,v_j)$ denote the capacity and cost of the edge joining vertices v_i and v_j respectively. Further, let $|A_n| = 3$ and $T_{max} = 2$ for the given scenario. Then we set $c(v_0,v_1) = 3$, $c(v_1,v_2) = 2$ and $c(v_1,v_3) = 2$. Also from Theorem 1, $c(v_2,v_x) = 1\ \forall x \in \{4,6\}$, $c(v_3,v_y) = 1\ \forall y \in \{6,9,11\}$ and $c(v_z,v_{13}) = 1\ \forall z \in \{4,5,6,7,8,9,10,11,12\}$. Also $co(v_k,v_{13}) = u_l(r_k)$. After reduction to the minimum cost maximum flow problem *MTA* and *MCMTA* can be solved using any of the standard techniques. We have used the Ford-Fulkerson algorithm followed by linear programming.

TABLE I
PARAMETER SETTING

Parameter	Value
Number of neighbouring SVs (N)	$\{2,3,4\}$
Maximum queue size of user (M^U_{max})	6
Maximum queue size of neighbouring SVs (M_{max})	3
Number of distance states (K)	3
Mean number of tasks processed by user per time slot (λ^U)	2
Mean number of tasks processed by other SVs per time slot (λ^i)	1
Maximum sensor data quality level (q_{max})	2
Application utility function constant (α)	10

8) Choosing $S(i)$: Sensor j of node i must be included in $S(i)$ if there is a flow on the edge between vertex v_{1+i} and $v_{1+|W|+\beta^i_j}$, where β^i_j denotes the FoV of sensor j of node i.

9) Choosing $|A_n|$: $|A_n|$ was defined in Definition 1 as the number of tasks that the user wants to process during a time slot t_n. Our approach for finding the optimum value of $|A_n|$ is to consider each value from 0 to $min(N^{max}_{Se}), N_{Ta}$. In each case we compute $S(i)\ \forall i \in \{1,2,...,N\}$ and evaluate $V(s)$. Then we choose the value for which $V(s)$ is minimum.

Since this approach involves both the maximum flow minimum cost based heuristic for sensor selection and the relative distance state based MDP framework, we will refer to this as the MDPH framework.

V. SIMULATION SETTING, RESULTS AND DISCUSSIONS

In this section, we evaluate the performance of our initial MDP framework versus the MDPH framework. We present the parameter values and the performance metrics used in our simulations and a brief discussion of the results.

A. Simulation Setting

Table I summarizes the values of the parameters used in our simulation. During simulation instance, the user and the neighbouring SVs move independently in a square area of size $500\text{m} \times 500\text{m}$ according to the Semi-Markov Smooth mobility model developed in [9]. The simulation was run for 10^6 time steps. We compare the performance of the policy obtained from the MDP framework to the policy obtained from the MDPH framework.

B. Performance Metrics

In each time slot, during the entire run of the simulation, a state and action pair will be obtained depending on the policy used. We define the mean reward ratio (MRR) as the average of the ratio of rewards from MDPH and MDP frameworks, during the entire run of the simulation. Mathematically it can be defined as in equation (14). A higher value of MRR indicates that the MDPH framework has achieved a performance close to that of the MDP framework.

$$MRR = \frac{1}{T}\sum_{i=1}^{T}\frac{R^t_H(s,a)}{R^t(s,a)} \tag{14}$$

Here $R^t_H(s,a)$ and $R^t(s,a)$ denote the rewards from the MDPH and the MDP frameworks respectively. T denotes the total duration of the simulation in terms of time steps.

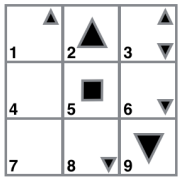
C. Results and Discussions

We vary $\bar{V}$, $\sigma_{\psi_{db}}$ and N, and plot the values of MRR defined in Section V-B, in each case. $\bar{V}$ is the average target speed in the SMS mobility model and $\sigma_{\psi_{db}}$ reprents the standard deviation (in db) of the path loss due to fading. Figures 2c, 2d and 2e plot the MRR versus $\bar{V}$, $\sigma_{\psi_{db}}$ and N respectively. Firstly, we observe that MRR decreases with an increase in $\bar{V}$, $\sigma_{\psi_{db}}$ and N. Before explaining this obsevation, we note that the MDP mainly models the uncertainty introduced by the V2V links, the available CPU speeds/queue space and node mobility. For a static network with infinite queue spaces (with all these uncertainties absent), the MDPH framework would perform exactly the same as the MDP framework. The MDP framework is likely to perform better in situations of higher uncertainty as it has more information to model this uncertainty compared to the MDPH framework. This is exactly what we observe in Figures 2c and 2d. With an increase in $\bar{V}$ and $\sigma_{\psi_{db}}$ the uncertainty due to node mobility and the V2V links respectively, increases. Thus, the MDP framework performs better and MRR drops. Figure 2f plots the MRR for two different choices of cost function used in the network flow graph. The curve named Ds plots the MRR of the MDPH framework with network flow graph cost function of $\frac{1}{u_l(l^i)}$, whereas the curve named Dqs plots the MRR for the MDPH framework with network flow graph cost function of $\frac{m^i}{u_l(l^i)}$. In the case of Dps the queue state information is built into the cost function. Therefore, there is a slight improvement in performance.

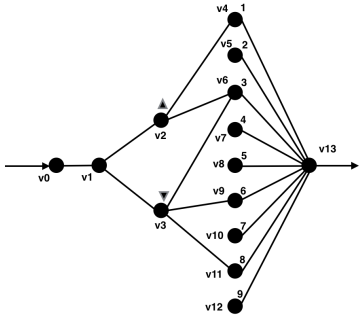
VI. RELATED WORKS

[10] proposes a quality of information (QoI) aware cooperative sensing scheme for VSNs. It considers a V2I scenario and is centralized. [11] proposes a reinforcement learning based trust mechanism and [12] proposes a deep reinforcement learning (DRL) based optimal sub-channel selection mechanism for a V2I scenario. Incorporating trust and sub-channel selection in our work will be interesting.

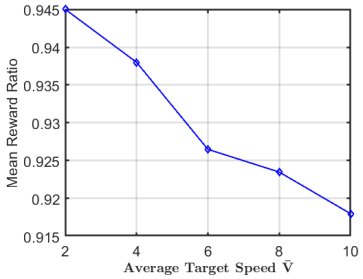
Cloud-enabled vehicular networks [5], mobile-edge computing (MEC) based offloading for vehicular networks [13] and vehicular fog computing (VFC) [14] are some paradigms proposed to overcome the resource limitation of SVs. However, the optimal resource allocation schemes were not considered. Some works in MANETs [15] have considered the use of MDPs for task offloading, but our work considers both sensing and task offloading jointly.



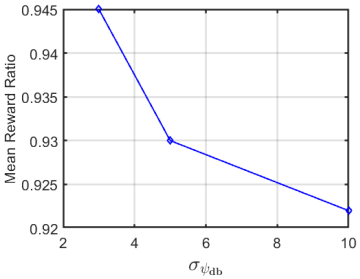
(a) An example scenario.



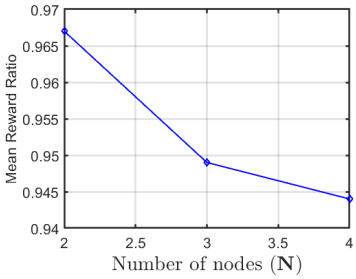
(b) Flow Network Graph.



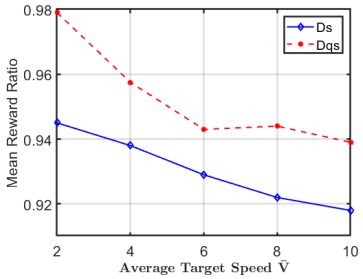
(c) Mean Reward Ratio vs Average Target Speed.



(d) Mean Reward Ratio vs $\sigma_{\psi_{db}}$.



(e) Mean Reward Ratio vs Number of Nodes.



(f) Mean Reward Ratio of the Ds and Dqs schemes vs Average Target Speed.

Fig. 2. Performance of MDP based scheme.

VII. Conclusion and Future Work

This paper presents a MDP based sensing and processing framework for VNs. The utility measure of a sensor requires information regarding its FoV. Hence, the state space must contain 2D location information and the framework does not scale well. We propose a maximum flow minimum cost based heuristic to convert the 2D location state to 1D relative distance state, making the framework scalable. Simulations show that the heuristic can obtain comparable performance.

VIII. Acknowledgment

This research is partially supported by AcRF, MOE, Singapore.

References

[1] Q. Fan, Y. Yi, L. Hao, F. Mengyin, and W. Shunting, "Semantic motion segmentation for urban dynamic scene understanding," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 497–502.

[2] A. Dewan, G. L. Oliveira, and W. Burgard, "Deep semantic classification for 3d LiDAR data," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 3544–3549.

[3] B. Wu, F. N. Iandola, P. H. Jin, and K. Keutzer, "SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving." in *CVPR Workshops*, 2017, pp. 446–454.

[4] D. Gschwend, "ZynqNet: An FPGA-accelerated embedded convolutional neural network," *vol. Master ETH-Zurich: Swiss Federal Institute of Technology Zurich*, 2016.

[5] R. Yu, X. Huang, J. Kang, J. Ding, S. Maharjan, S. Gjessing, and Y. Zhang, "Cooperative resource management in cloud-enabled vehicular networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7938–7951, 2015.

[6] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical Systems and Signal Processing*, vol. 100, pp. 482–500, 2018.

[7] M. Riahi, T. G. Papaioannou, I. Trummer, and K. Aberer, "Utility-driven data acquisition in participatory sensing," in *Proceedings of the 16th International Conference on Extending Database Technology*. ACM, 2013, pp. 251–262.

[8] M. Zhao, Y. Li, and W. Wang, "Modeling and analytical study of link properties in multihop wireless networks," *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 445–455, 2012.

[9] M. Zhao and W. Wang, "A unified mobility model for analysis and simulation of mobile wireless networks," *Wireless Networks*, vol. 15, no. 3, pp. 365–389, 2009.

[10] D. Van Le, C.-K. Tham, and Y. Zhu, "Quality of information (QoI)-aware cooperative sensing in vehicular sensor networks," in *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 369–374.

[11] J. Wang, C. Jiang, K. Zhang, T. Q. Quek, Y. Ren, and L. Hanzo, "Vehicular sensing networks in a smart city: Principles, technologies and applications," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 122–132, 2018.

[12] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for v2v communications," *arXiv preprint arXiv:1805.07222*, 2018.

[13] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, 2017.

[14] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[15] D. Van Le and C.-K. Tham, "Quality of service aware computation offloading in an ad-hoc mobile cloud," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8890–8904, 2018.