## Emergency Awareness Extension

### Extended problem

This problem is an extension of the traditional traffic congestion control problem to incorporate emergency-awareness. The system is formulated mathematically as follows.

We consider an urban environment comprising multiple junctions, where each junction represents an intersection between horizontal and vertical lines. Each line consists of set of roads with two directions, and each direction has multiple lanes. Vehicles generation occurs at all roads where they intersect with the environment border in the IN, and vehicle departure occurs in the out direction at the same intersections.

Each junction operates in two modes: road-centered and junction-centered. In the road-centered mode, actions enable traffic flow either east and west (EW) or north and south (S-N) (left permitted). In the junction centered mode, actions enable the opening one of the four roads (E, W, N and S) to three directions (left protected) to three directions, namely, forward, left, and right. An additional “emergency-aware mode” is introduced, which prioritizes the passage of emergency vehicles through the junctions by dynamically adjusting the actions and states based on real-time emergency data.

We assume that each road is equipped with load cells at its two ends for measuring the flow of vehicles and cameras for counting vehicles. Additionally, vehicles are equipped with IEEE 802.11p for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication. Roadside units are available at each junction for facilitating this communication.

Given that the vehicles are generated using a normal distribution probability density function for several vehicles and an exponential distribution probability density function for time intervals, and considering that the sensors are subject to noise, our goal is twofold:

1. To exploit sensing, V2V, and V2I communication for state estimation using sensor fusion.
2. To control traffic using reinforcement learning, with an added focus on emergency-aware traffic management.

### RL based solution

The agent which currently operates at the base-station, could be extended to include emergency aware module. This module would be responsible for identifying emergency situations, such as ambulances or fire trucks requiring immediate passage. The agent could receive real-time data from emergency services or specialized sensors to recognize such scenarios.

#### Introducing new actions

To make the system emergency-aware, new actions could be introduced into the existing set . These could include actions like “clear path for emergency” (CPE) and resume normal operation (RNO). The extended set of actions is .

#### Modifying state

The state space will be extended to include emergency conditions. For instance, each of the 16 states could have an additional binary flag indicating the presence or absence of an emergency. This would effectively double the state space to 32 states.

Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, facilitated through IEEE 802.11p, will be exploited to relay emergency information. Emergency vehicles could broadcast their status and location to nearby vehicles and roadside units, which in turn update the state information at the base station where the RL agent operates. This real-time data will be crucial for the RL agent to make informed decisions, particularly when an emergency vehicle is approaching a junction.

#### Adaptive reward function

The reward function will be modified to include penalties for not adequately responding to emergencies. For example, a large negative reward will be given if the algorithm fails to clear path for an emergency vehicle within certain time frame.

|  |  |
| --- | --- |
|  | (‎0.35) |

Where:

denotes the average waiting time of regular vehicles

denotes the average waiting time of emergency vehicles

denotes the number of times the system fails to prioritize emergency vehicles within certain time frame

, , denotes waiting factors that can be tuned to prioritize one term over the others

The first term aims to minimize the average waiting time for regular traffic, while the second term focuses on minimizing the waiting time for emergency vehicles. The third term penalizes the system for not adequately responding to emergency situations.

#### Optimization of Infrastructure Structure

Let be an urban environment defined by set of roads , junctions , and a set of emergency routes . Let be the set of candidate locations for within that will be responsible on supporting emergency routes. Our goal is to find subset of for deploying with the goal of optimizing various metrics, namely, coverage, cost, latency, and reliability.

Let be a binary vector of length where =1 If an RSU is deployed at location and =0

Otherwise.

|  |  |
| --- | --- |
|  | (‎0.36) |

Where:

denotes the pareto front

denotes the coverage achieved by the newly installed roadside unit

denotes the cost of installing the roadside units and deploying them

denotes the expected latency of packets after installing the roadside units

denotes the reliability of the communication that is achieved by installing and deploying the roadside units

1. Coverage

The coverage is defined as the ratio of the area that is covered by the deployed RSUs to the total area requiring coverage. Assuming that each RSU has a coverage radius , the coverage can be mathematically expressed as

|  |  |
| --- | --- |
|  | (‎0.37) |

1. Cost objective

The cost function includes both the capital expense CAPEX and operation expense OPEX for deploying and maintaining RSUs. Mathematically, this can be represented as

|  |  |
| --- | --- |
|  | (‎0.38) |

Where:

denotes the installation cost at location

denotes the operational cost at location

1. Latency

Latency is modeled as a function of distance between RSUs and nearest data center or aggregated point, as well as the processing time at the RSUs. For simplicity, we consider just the distance factor. For each RSU , we compute the expected latency as

|  |  |
| --- | --- |
|  | (‎0.39) |

1. Reliability

Reliability is either modeled as the probability that a transmitted packet is successfully received. This probability includes signal to noise ratio (SNR). It is given as

|  |  |
| --- | --- |
|  | (‎0.40) |

Where

denotes the SNR at each location

The reliability of the wireless communication link with a certain SNR is represented through bit error rate (BER) which is approximated for a binary phase-shift keying (BPSK) modulation scheme as

|  |  |
| --- | --- |
|  | (‎0.41) |
|  | (‎0.42) |
|  | (‎0.43) |

denotes the Q-function

##### NSGA-II Optimization Algorithm

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is an evolutionary algorithm designed for multi-objective optimization problems. It was developed by Kalyanmoy Deb in 2002 as an improvement over its predecessor, NSGA, with the primary objective of addressing various limitations in handling a wide range of optimization problems.

Let us consider a multi-objective optimization problem with objectives The optimization problem is formulated as

|  |  |
| --- | --- |
|  | (‎0.44) |

Where: is the feasible search space

A solution is said to dominate another solution , if:

1. Initialization

Generate an initial population of size

Set of generation counter

1. Evaluate of the objective function for each individual in
2. Perform non-dominated sorting on to classify individuals into different fronts …
3. For each front , calculate the crowding distance for each individual
4. Conduct binary tournament selection based on rank and crowding distance to create a mating pool
5. Apply crossover and mutation operators to the meeting pool to produce an offspring population of size
6. Combine the parents and offspring population
7. Perform non-dominated sorting on and select the best individuals based on rank and crowding distance to create
8. If the stopping criterion is met, terminate the algorithm. Otherwise, set and go to Step 2.

The pseudocode of the algorithm is presented in Algorithm 3.1.

Algorithm ‎0.1 Pseudocode of non-dominated sorting genetic algorithm

|  |
| --- |
| **Input**:  N  Objectives  **Output**:  in  **Start**:  1. Initialize population of size N  2. Evaluate objectives of  3.  4. WHILE (not termination condition)  4.1 Perform non-dominated sorting on  4.2 Calculate crowding distance for each front in  4.3 Perform binary tournament selection, crossover, and mutation to create  4.4 = U  4.5 Perform non-dominated sorting on  4.6 Create +1 by selecting the best individuals from based on rank and crowding distance  4.7  5. END WHILE |

## Evaluation

This section provides the evaluation design and the performance metrics used for evaluating our objectives and comparing them with benchmark. It is decomposed of two sub-sections. First, we present the evaluation metrics in sub-section – and next we present the experimental design in sub-section --.

### Evaluation metrics

For evaluating our approach, we use three evaluation metrics, namely, the estimation error of has the role of evaluating the estimation, the average waiting time, and the number of released vehicles, which are responsible for measuring the performance of the traffic control.

The evaluation will use the following metrics:

#### Error

The estimation error for each road is and it represents the difference in the number of vehicles between the true ones and the estimated ones by the Kalman Filtering approach. This metric needs to be minimized.

|  |  |
| --- | --- |
|  | (‎0.45) |

Where:

denotes the number of vehicles in the road

denotes the estimated nu

#### Average Waiting Time

It represents the average waiting time of the vehicles in the road of the junction before they are released.

|  |  |
| --- | --- |
|  | (‎0.46) |

Where:

denotes the waiting time of vehicle in the road

#### Fairness

The fairness is measured by the standard deviation of the waiting time between the different roads. It is given by the Equation (3.47).

|  |  |
| --- | --- |
|  | (‎0.47) |