Fuzzy Logic Solution for Calculation of Emergency Response Plan

ERP System

The Emergency Response Plan is decided by the Congestion Level and Emergency level, several ERP's exist and one is chosen by the combination of these two inputs. Emergency level is decided by the emergency response services (for now I will assume only three EL's exist High, Medium, Low, given values 3, 2, 1) and the Congestion level is calculated through fuzzy logic. Some explanation of Emergency and Congestion Levels is given below.

Emergency Levels:

Emergency levels denote the importance of a fast arrival by the emergency vehicle, higher emergency levels permit the adaptive system to de-prioritise Congestion management to some extent.

Class	Cases	EL
Low	Minor Injuries, Disturbances, Low risk of property damage.	1
Moderate	Serious but non-life threatening injuries or chance of moderate damage.	2
High	Potential loss of human life or excessive property damage.	3

Congestion level

The Congestion Level (CL) is derived from the combination of three inputs which gives a Congestion Value (C) which is mapped to a CL as per this table:

Congestion Value	Class	CL
$C <= 0.1$	Minimal	1
$0.1 < C <= 0.25$	Low	2
$0.25 < C <= 0.5$	Moderate	3
$0.5 < C <= 0.7$	High	4
$0.7 < C$	Critical	5

More on this in the next section.

ERP Types and Changes

Emergency response plans can administer different changes to driving policies depending on the congestion and emergency levels. First we will list the changes it can make and then we will define a few ERPs.

Changes

Traffic Lights: Ensure the Vehicle only meets green lights.
Estimated Congestion Impact: Low
Identifier: **TL**

Speed Limit: Increase speed limit temporarily.
Estimated Congestion Impact: Low
Identifier: **SL**

Lane Clearance: Clear entire lanes for the vehicle.
Estimated Congestion Impact: High
Identifier: **LC**

Reserved lanes: Allow vehicles to use reserved lanes (bus tax etc.)
Estimated Congestion Impact: Moderate
Identifier: **RL**

Turning-lanes: Extension of RL to include turning lanes.
Identifier: **TLA**

Re-routing: Re-route vehicles away from route.
Estimated Congestion Impact: Depends on implementation.
Identifier: **RR**

<u>Plans:</u>
ERP 1: Lowe-medium traffic and not too much severity. Simplest case.
TL

ERP 2: Lower-medium traffic, some severity.
TL, SL

ERP 3 : Lower-medium Traffic high severity , we care about getting there fast more than traffic
TL, SL, LC, RL

ERP 4 : High Traffic Low-Medium Severity, must consider impact on traffic though of course the emergency takes priority.
TL, SL, LC, RR

ERP 5 : High traffic high severity. Worst case scenario, use everything at our disposal.
TL, SL, LC, RL, TLA, RR

CL	EL	Suggested ERP
Minimal	Low	1
Minimal	Medium	1
Minimal	High	3
Low	Low	1
Low	Medium	2
Low	High	3
Medium	Low	1
Medium	Medium	2
Medium	High	3
High	Low	4
High	Medium	4
High	High	5
Critical	Low	4
Critical	Medium	5
Critical	High	5

Congestion Level

Congestion Level (CL) is calculated by the combination of three weighted inputs: 1) Occupancy Level (OL) which represents the percentage occupancy of the edges on the Emergency Vehicle's Route (EVR), this is a value between 0 and 1 which is adjusted with a fuzzy logic solution. 2) Average Vehicle Speed (AVS) represents the average speed of vehicles on the edges of the EVR. Again to be adjusted with Fuzzy logic. 3) Time Based Demand Estimate (TBDE) value corresponding to the time of day/week (rush hour etc.) to consider the likelihood of traffic increase or decrease.

All three inputs will have a value between 0 and 1 and be multiplied by their respective weights (I suggest a weight of 0.6 for OL, 0.3 for AVS and 0.1 for TBDE) to give the final congestion value.
So with the suggested weights C = 0.6*OL + 0.3*AVS + 0.1*TBDE. C is mapped to CL as per the table in the first section.

<u>Occupancy Level:</u>

The initial occupation value λ is returned from SUMO (or from traffic monitoring sensors in a real application) and then adjusted with fuzzy logic to give an output OL value. The system aims to map the value to one or more of five fuzzy classes {Very Low, Low, Moderate, High, Very High} (let's call them {a, b, c, d, e} for now) each with its own **centre c(i)** where *i* is a subset {a,b,c,d,e}, each fuzzy class has a lower and upper boundary as well where boundaries of one class may overlap the boundaries of another.

The fuzzy OL value returned will also be a value between one and zero where equal to the sum of c(i)*m(OL, i) where I = {a,b,c,d,e} and **m(λ, i) is the membership** of the input λ. The sum of m(λ, i) should always be equal to 1 (since a level can belong for example 60% to one class and 40% to another) .

<u>AVS:</u>

Calculated from an initial average speed value α (again taken from SUMO) and adjusted by fuzzy logic in the same way as for the OL value. The main difference is that there may be different class names or numbers (I would suggest 5 for now: very Slow, Slow Medium, Fast, Dangerously fast). The membership classes will be less evenly distributed here since their boundaries will not be In the range 0-1 but 0-X where X is the highest possible average speed, since there will be a large amount of high speeds that are very unlikely the 'dangerously fast' class will span a large range of values (let's say 90km/h – X km/h). The fuzzy logic function will normalise the value to give an AVS between 0 and 1.

<u>TBDE:</u>

Enumeration of values 0-1.
Suggested:
0.2 = Very Low Demand (between 0:00 and 4:00 on weekdays)
0.4: Low demand (e.g. : 12:00 – 15:00 on weekends in winter)
Moderate Demand: = 0.6 (e.g. : lunchtime on weekdays)
High Demand = 0.8 (e.g. : Rush hour)
Should be set locally based on city/town/area, population demographics and traffic stats.