References

[1] "Statistical Pocket Book 2016," Department of Census and Statistics - Sri Lanka, Survey.

[2] C. Xiao-Feng, S. Zhong-ke, and Z. Kai, "Research on an intelligent traffic signal controller," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, 2003, vol. 1, pp. 884–887.

[3] C. Li and S. Shimamoto, "A real time traffic light control scheme for reducing vehicles CO 2 emissions," in *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, 2011, pp. 855–859.

[4] N. J. Garber and L. A. Hoel, *Traffic and highway engineering*. Cengage Learning, 2014.

[5] C. Chen, Z. Jia, and P. Varaiya, "Causes and cures of highway congestion," *IEEE Control Syst. Mag.*, vol. 21, no. 6, pp. 26–32, 2001.

[6] M. Litzenberger *et al.*, "Vehicle counting with an embedded traffic data system using an optical transient sensor," in *2007 IEEE Intelligent Transportation Systems Conference*, 2007, pp. 36–40.

[7] Bo Chen and H. H. Cheng, "A Review of the Applications of Agent Technology in Traffic and Transportation Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 485–497, Jun. 2010.

[8] K. Singh and B. Li, "Estimation of Traffic Densities for Multilane Roadways Using a Markov Model Approach," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4369–4376, Nov. 2012.

[9]Z. Tehao and C. Feng, "Acquisition of traffic flow density using multi-source data fusion," in *Measurement, Information and Control (MIC), 2012 International Conference on*, 2012, vol. 2, pp. 595–599.

[10] J. Barrachina *et al.*, "A V2I-Based Real-Time Traffic Density Estimation System in Urban Scenarios," *Wirel. Pers. Commun.*, vol. 83, no. 1, pp. 259–280, Jul. 2015.

[11] R. Mao and G. Mao, "Road traffic density estimation in vehicular networks," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, 2013, pp. 4653–4658.

[12] L. Garelli, C. Casetti, C.-F. Chiasserini, and M. Fiore, "Mobsampling: V2V communications for traffic density estimation," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, 2011, pp. 1–5.

[13] T. Darwish and K. Abu Bakar, "Traffic density estimation in vehicular ad hoc networks: A review," *Ad Hoc Netw.*, vol. 24, pp. 337–351, Jan. 2015.

[14] A. Tabibiazar and O. Basir, "Kernel-based modeling and optimization for density estimation in transportation systems using floating car data," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011, pp. 576–581.

[15] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular Traffic Density State Estimation Based on Cumulative Road Acoustics," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1156–1166, Sep. 2012.

[16] M. M. nal Bhandarkar and M. T. Waykole, "Traffic Density Estimation and Mechanical Condition Determination of Vehicles using Acoustic Signals."

[17] P. Borkar and L. G. Malik, "Acoustic signal based traffic density state estimation using adaptive Neuro-Fuzzy classifier," in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, 2013, pp. 1–8.

[18] J. Kato, Y. Hiramatsu, and T. Watanabe, "Estimating traffic density using sounds of moving vehicles," in *Proceedings of the Seventh Eurographics conference on Multimedia*, 2004, pp. 21–29.

[19] J. Kato, “An attempt to acquire traffic density by using road traffic sound,” in *Active Media Technology, 2005.(AMT 2005). Proceedings of the 2005 International Conference on*, 2005, pp. 353–358.
[20] E. Tan and J. Chen, “Vehicular traffic density estimation via statistical methods with automated state learning,” in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, 2007, pp. 164–169.
[21] T. Wassantachat, Z. Li, J. Chen, Y. Wang, and E. Tan, “Traffic Density Estimation with On-line SVM Classifier,” 2009, pp. 13–18.
[22] S. B. Purusothaman and K. Parasuraman, “Vehicular traffic density state estimation using Support Vector Machine,” in *Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013 International Conference on*, 2013, pp. 782–785.
[23] Z. Li, E. Tan, J. Chen, and T. Wassantachat, “On Traffic Density Estimation with a Boosted SVM Classifier,” 2008, pp. 117–123.
[24] P. Janney and G. Geers, “Advanced framework for illumination invariant traffic density estimation,” in *Intelligent Transportation Systems, 2009. ITSC’09. 12th International IEEE Conference on*, 2009, pp. 1–6.
[25] O. Asmaa, K. Mokhtar, and O. Abdelaziz, “Road traffic density estimation using microscopic and macroscopic parameters,” *Image Vis. Comput.*, vol. 31, no. 11, pp. 887–894, Nov. 2013.
[26] Y. Yuan, L. Mou, and X. Lu, “Scene Recognition by Manifold Regularized Deep Learning Architecture,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–1, 2015.
[27] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[28] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, “Learning deep features for scene recognition using places database,” in *Advances in Neural Information Processing Systems*, 2014, pp. 487–495.
[29] S. Mehan and V. Sharma, “Development of traffic light control system based on fuzzy logic,” in *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence*, 2011, pp. 162–165.
[30] K. T. K. Teo, K. B. Yeo, S. E. Tan, Z. W. Siew, and K. G. Lim, “Design and development of portable Fuzzy Logic based traffic optimizer,” in *ICCE-China Workshop (ICCE-China), 2013 IEEE*, 2013, pp. 7–12.
[31] K. T. K. Teo, W. Y. Kow, and Y. K. Chin, “Optimization of Traffic Flow within an Urban Traffic Light Intersection with Genetic Algorithm,” 2010, pp. 172–177.
[32] F. Teklu, A. Sumalee, and D. Watling, “A genetic algorithm approach for optimizing traffic control signals considering routing,” *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 22, no. 1, pp. 31–43, 2007.
[33] Y. K. Chin, W. Y. Kow, W. L. Khong, M. K. Tan, and K. T. K. Teo, “Q-Learning Traffic Signal Optimization within Multiple Intersections Traffic Network,” 2012, pp. 343–348.
[34] A. W. Addison, T. N. Rao, J. Reedijk, J. van Rijn, and G. C. Verschoor, “Synthesis, structure, and spectroscopic properties of copper (II) compounds containing nitrogen–sulphur donor ligands; the crystal and molecular structure of aqua [1, 7-bis (N-methylbenzimidazol-2′-yl)-2, 6-dithiaheptane] copper (II) perchlorate,” *J. Chem. Soc. Dalton Trans.*, no. 7, pp. 1349–1356, 1984.
[35] W. Narzt, U. Wilflingseder, G. Pomberger, D. Kolb, and H. Hörtner, “Self-organising congestion evasion strategies using ant-based pheromones,” *IET Intell. Transp. Syst.*, vol. 4, no. 1, p. 93, 2010.

[36] D. Renfrew and X.-H. Yu, “Traffic Signal Control with Swarm Intelligence,” in *Proceedings of the 2009 Fifth International Conference on Natural Computation - Volume 03*, Washington, DC, USA, 2009, pp. 79–83.
[37] D. Renfrew and X.-H. Yu, “Traffic Signal Control with Swarm Intelligence,” 2009, pp. 79–83.
[38] J. García-Nieto, E. Alba, and A. C. Olivera, “Enhancing the urban road traffic with Swarm Intelligence: A case study of Córdoba city downtown,” in *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, 2011, pp. 368–373.
[39] J. Zhao and D. Tang, “Coordination traffic control under the framework of multi-agent technology,” in *Computer Science and Engineering, 2009. WCSE’09. Second International Workshop on*, 2009, vol. 1, pp. 219–222.
[40] J. Zhang, L. Yan, Y. Han, G. Song, and X.-L. Fang, “Research on the Method and Simulation of Intersection Signal Control Based on Multi-Agent,” in *Management and Service Science, 2009. MASS’09. International Conference on*, 2009, pp. 1–4.
[41] W. Wu, G. Haifei, and J. An, “A Multi-agent Traffic Signal Control System Using Reinforcement Learning,” in *2009 Fifth International Conference on Natural Computation*, 2009, vol. 4, pp. 553–557.
[42] A. Krogh, “What are artificial neural networks?,” *Nat. Biotechnol.*, vol. 26, no. 2, pp. 195–197, 2008.
[43] T. Dean, “A computational model of the cerebral cortex,” in *Proceedings of the National Conference on Artificial Intelligence*, 2005, vol. 20, p. 938.
[44] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Robust object recognition with cortex-like mechanisms,” *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 29, no. 3, pp. 411–426, 2007.
[45] I. Arel, D. C. Rose, and T. P. Karnowski, “Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier],” *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 13–18, Nov. 2010.
[46] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks.,” in *Aistats*, 2011, vol. 15, p. 275.
[47] “http://caffe.berkeleyvision.org/.” .
[48] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Englewood Cliffs, N.J: Prentice Hall, 1995.
[49] G. Weiss, Ed., *Multiagent systems: a modern approach to distributed artificial intelligence*. Cambridge, Mass: MIT Press, 1999.
[50] V. I. Gorodetskii, “Self-organization and multiagent systems: I. Models of multiagent self-organization,” *J. Comput. Syst. Sci. Int.*, vol. 51, no. 2, pp. 256–281, Apr. 2012.
[51] M.-P. Huget, *Communication in Multiagent Systems: Agent communication languages and conversation policies*, vol. 2650. Springer Science & Business Media, 2003.
[52] Y. Jia *et al.*, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.
[53] “https://ffmpeg.org/ffmpeg.html.” .
[54] “http://scikit-image.org/.” .
[55] I. J. Timóteo, M. R. Araújo, R. J. Rossetti, and E. C. Oliveira, “TraSMAPI: An API oriented towards Multi-Agent Systems real-time interaction with multiple Traffic Simulators,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 2010, pp. 1183–1188.

Appendix A

Road traffic scenes of different intersections

Figure A.1: Narahenpita Junction

Figure A.2: Ayurveda Junction

Figure A.3: Kanaththa Junction

Appendix B

Class diagram for traffic control optimization

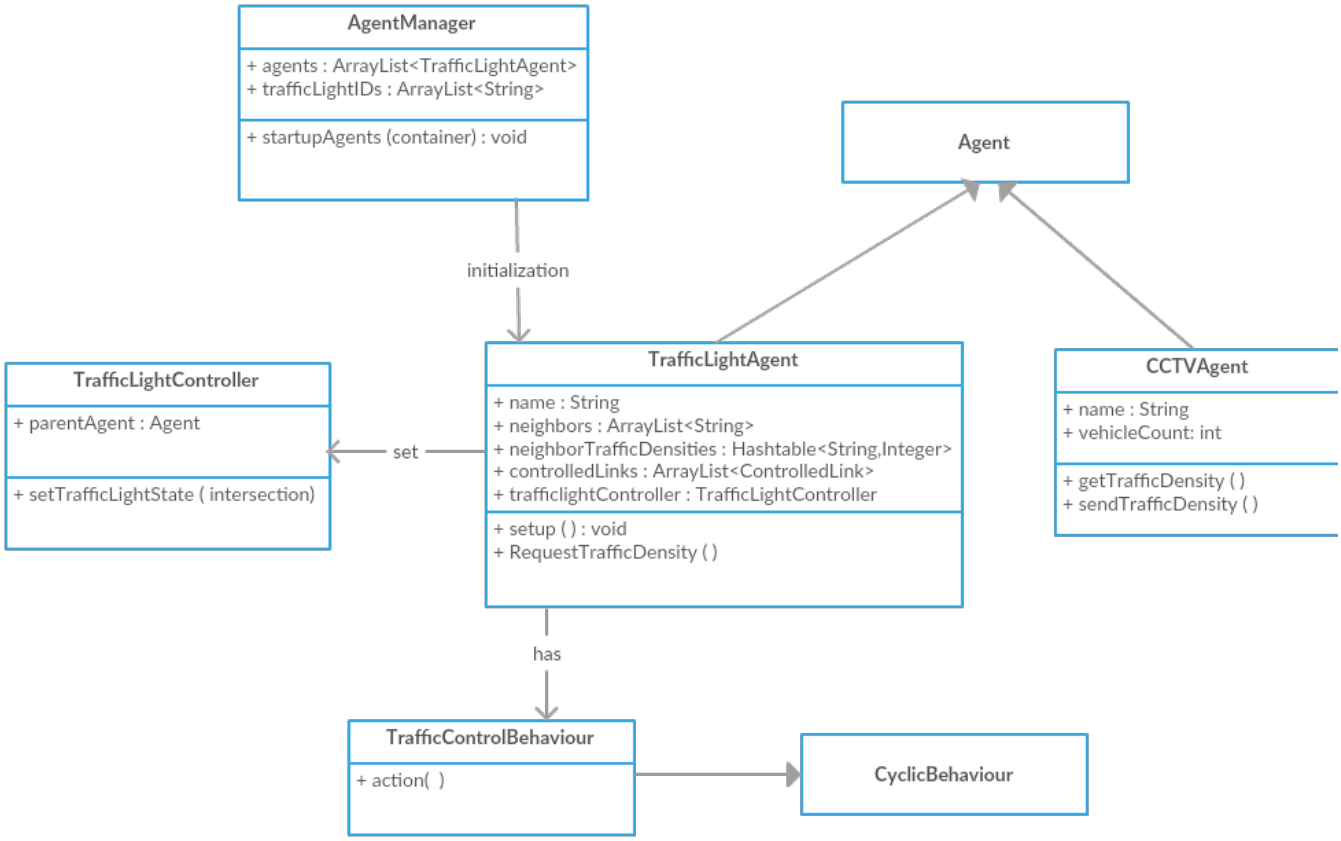

Figure B.1: Class diagram for traffic control optimization

Appendix C

Configuration details in Caffe Solver

Traffic_solver.prototxt

```
# The train/test net protocol buffer definition
net: "/Users/jwithanawasam/MachineLearning/Caffe/caffe/Traffic/traffic_train_test.prototxt"
# test_iter specifies how many forward passes the test should carry out.
# In the case of MNIST, we have test batch size 100 and 100 test iterations,
# covering the full 10,000 testing images.
test_iter: 100
# Carry out testing every 500 training iterations.
#test_interval: 500
test_interval: 500
# The base learning rate, momentum and the weight decay of the network.
#base_lr: 0.01
base_lr: 0.00009
momentum: 0.9
weight_decay: 0.0005
# The learning rate policy
lr_policy: "inv"
gamma: 0.0001
power: 0.75
# Display every 100 iterations
display: 100
# The maximum number of iterations
max_iter: 10000
# snapshot intermediate results
snapshot: 5000
snapshot_prefix: "/Users/jwithanawasam/MachineLearning/Caffe/caffe/Traffic/model"
# solver mode: CPU or GPU
solver_mode: CPU
```

Appendix D

Configuration details for Caffe deep neural network

Train_test.prototxt

```
name: "LeNet"
layer {
  name: "mnist"
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TRAIN
  }
  transform_param {
    scale: 0.00390625
  }
  data_param {
    source: "/Users/jwithanawasam/MachineLearning/Caffe/caffe/Traffic/training/train_lmdb"
    batch_size: 16
    backend: LMDB
  }
}
layer {
  name: "mnist"
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TEST
  }
  transform_param {
    scale: 0.00390625
```

```
  }
  data_param {
    source: "/Users/jwithanawasam/MachineLearning/Caffe/caffe/Traffic/testing/test_lmdb"
    batch_size: 16
    backend: LMDB
  }
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "conv1"
```

```
  top: "pool1"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
  }
}
layer {
  name: "conv2"
  type: "Convolution"
  bottom: "pool1"
  top: "conv2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 50
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "pool2"
  type: "Pooling"
  bottom: "conv2"
```

```
  top: "pool2"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
  }
}
layer {
  name: "ip1"
  type: "InnerProduct"
  bottom: "pool2"
  top: "ip1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 500
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "ip1"
  top: "ip1"
}
```

```
layer {
  name: "ip2"
  type: "InnerProduct"
  bottom: "ip1"
  top: "ip2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 2
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "accuracy"
  type: "Accuracy"
  bottom: "ip2"
  bottom: "label"
  top: "accuracy"
  include {
    phase: TEST
  }
}
layer {
  name: "loss"
  type: "SoftmaxWithLoss"
```

bottom: "ip2"
bottom: "label"
top: "loss" }

Appendix E

Image Classification using PyCaffe

```
import numpy as np
import matplotlib.pyplot as plt
import caffe

DEPLOY_LOCATION =
'/Users/jwithanawasam/MachineLearning/Caffe/caffe/Traffic/traffic_deploy.prototxt'
MODEL_LOCATION =
'/Users/jwithanawasam/MachineLearning/Caffe/caffe/Traffic/model_iter_10000.caffemodel'
IMAGE_LOCATION = '/Users/jwithanawasam/MachineLearning/Caffe/caffe/Traffic/
/images/traffic_img_00001.jpeg'

caffe.set_mode_cpu()
net = caffe.Classifier(DEPLOY_LOCATION, MODEL_LOCATION)
input_image = caffe.io.load_image(IMAGE_LOCATION)
plt.imshow(input_image)
prediction = net.predict([input_image])
plt.plot(prediction[0])
print 'Traffic density state:', prediction[0].argmax()
plt.show()
```

Appendix F

Road traffic simulation using SUMO

Node definition

```
<nodes version="0.13" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/nodes_file.xsd">
    <location netOffset="0.00,0.00" convBoundary="0.00,0.00,300.00,300.00"
origBoundary="0.00,0.00,300.00,300.00" projParameter="!"/>

    <node id="0/0" x="0.00" y="0.00" type="priority"/>
    <node id="0/1" x="0.00" y="150.00" type="traffic_light" tl="0/1"/>
    <node id="0/2" x="0.00" y="300.00" type="priority"/>
    <node id="1/0" x="150.00" y="0.00" type="traffic_light" tl="1/0"/>
    <node id="1/1" x="150.00" y="150.00" type="traffic_light" tl="1/1"/>
    <node id="1/2" x="150.00" y="300.00" type="traffic_light" tl="1/2"/>
    <node id="2/0" x="300.00" y="0.00" type="priority"/>
    <node id="2/1" x="300.00" y="150.00" type="traffic_light" tl="2/1"/>
    <node id="2/2" x="300.00" y="300.00" type="priority"/>
</nodes>
```

Edges definition

```
<edges version="0.13" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://
sumo-sim.org/xsd/edges_file.xsd">
    <edge id="0/0to0/1" from="0/0" to="0/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="0/0to1/0" from="0/0" to="1/0" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="0/1to0/0" from="0/1" to="0/0" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="0/1to0/2" from="0/1" to="0/2" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="0/1to1/1" from="0/1" to="1/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="0/2to0/1" from="0/2" to="0/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="0/2to1/2" from="0/2" to="1/2" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/0to0/0" from="1/0" to="0/0" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/0to1/1" from="1/0" to="1/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/0to2/0" from="1/0" to="2/0" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/1to0/1" from="1/1" to="0/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/1to1/0" from="1/1" to="1/0" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/1to1/2" from="1/1" to="1/2" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/1to2/1" from="1/1" to="2/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/2to0/2" from="1/2" to="0/2" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/2to1/1" from="1/2" to="1/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="1/2to2/2" from="1/2" to="2/2" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="2/0to1/0" from="2/0" to="1/0" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="2/0to2/1" from="2/0" to="2/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="2/1to1/1" from="2/1" to="1/1" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="2/1to2/0" from="2/1" to="2/0" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="2/1to2/2" from="2/1" to="2/2" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="2/2to1/2" from="2/2" to="1/2" priority="-1" numLanes="1" speed="13.90"/>
    <edge id="2/2to2/1" from="2/2" to="2/1" priority="-1" numLanes="1" speed="13.90"/>
</edges>
```

Connections definition

```
<connections version="0.13" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/connections_file.xsd">

    <connection from="0/0to0/1" to="0/1to1/1" fromLane="0" toLane="0"/>
    <connection from="0/0to0/1" to="0/1to0/2" fromLane="0" toLane="0"/>

    <connection from="0/0to1/0" to="1/0to2/0" fromLane="0" toLane="0"/>
    <connection from="0/0to1/0" to="1/0to1/1" fromLane="0" toLane="0"/>

    <connection from="0/1to0/0" to="0/0to1/0" fromLane="0" toLane="0"/>

    <connection from="0/1to0/2" to="0/2to1/2" fromLane="0" toLane="0"/>

    <connection from="0/1to1/1" to="1/1to1/0" fromLane="0" toLane="0"/>
    <connection from="0/1to1/1" to="1/1to2/1" fromLane="0" toLane="0"/>
    <connection from="0/1to1/1" to="1/1to1/2" fromLane="0" toLane="0"/>

    <connection from="0/2to0/1" to="0/1to0/0" fromLane="0" toLane="0"/>
    <connection from="0/2to0/1" to="0/1to1/1" fromLane="0" toLane="0"/>

    <connection from="0/2to1/2" to="1/2to1/1" fromLane="0" toLane="0"/>
    <connection from="0/2to1/2" to="1/2to2/2" fromLane="0" toLane="0"/>

    <connection from="1/0to0/0" to="0/0to0/1" fromLane="0" toLane="0"/>

    <connection from="1/0to1/1" to="1/1to2/1" fromLane="0" toLane="0"/>
    <connection from="1/0to1/1" to="1/1to1/2" fromLane="0" toLane="0"/>
    <connection from="1/0to1/1" to="1/1to0/1" fromLane="0" toLane="0"/>

    <connection from="1/0to2/0" to="2/0to2/1" fromLane="0" toLane="0"/>
</connections>
```

Routes definition for vehicles

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/routes_file.xsd"><vType accel="1.8" color="red" decel="4.5"
id="eme" length="5" maxSpeed="140" sigma="0.5" vClass="emergency"/><vType accel="0.4" color="blue" decel="8.5"
id="bus" length="10" maxSpeed="40" sigma="0.5" vClass="bus"/><vType accel="2.0" decel="4.5" id="nor" length="5"
maxSpeed="120" sigma="0.5"/><vehicle depart="1.00" id="0" type="nor">
        <route edges="2/2to2/1 2/1to1/1 1/1to1/2 1/2to2/2"/>
    </vehicle><vehicle depart="5.00" id="1" type="eme">
        <route edges="2/2to1/2 1/2to1/1 1/1to0/1 0/1to0/2 0/2to1/2 1/2to2/2 2/2to2/1"/>
    </vehicle><vehicle depart="9.00" id="2" type="eme">
        <route edges="2/1to2/2 2/2to1/2 1/2to1/1 1/1to0/1 0/1to0/0 0/0to1/0 1/0to1/1"/>
    </vehicle><vehicle depart="13.00" id="3" type="eme">
        <route edges="1/1to2/1 2/1to2/0 2/0to1/0 1/0to1/1 1/1to1/2 1/2to2/2 2/2to2/1"/>
    </vehicle><vehicle depart="17.00" id="4" type="eme">
        <route edges="0/2to0/1 0/1to1/1 1/1to1/2"/>
    </vehicle><vehicle depart="21.00" id="5" type="eme">
        <route edges="0/2to1/2 1/2to1/1 1/1to1/0 1/0to0/0 0/0to0/1"/>
    </vehicle><vehicle depart="25.00" id="6" type="eme">
        <route edges="2/0to2/1 2/1to2/2 2/2to1/2"/>
</vehicle></routes>
```

Traffic signal phase and cycle definition

```
<tlLogics version="0.13" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo-sim.org/xsd/tllogic_file.xsd">
    <tlLogic id="0/1" type="static" programID="0" offset="0">
        <phase duration="31" state="GgrrGG"/>
        <phase duration="4" state="ygrryy"/>
        <phase duration="6" state="rGrrrr"/>
        <phase duration="4" state="ryrrrr"/>
        <phase duration="31" state="rrGGGr"/>
        <phase duration="4" state="rryyyr"/>
    </tlLogic>
    <tlLogic id="1/0" type="static" programID="0" offset="0">
        <phase duration="31" state="rrGGGg"/>
        <phase duration="4" state="rryyyg"/>
        <phase duration="6" state="rrrrrG"/>
        <phase duration="4" state="rrrrry"/>
        <phase duration="31" state="GGGrrr"/>
        <phase duration="4" state="yyyrrr"/>
    </tlLogic>
    <tlLogic id="1/1" type="static" programID="0" offset="0">
        <phase duration="31" state="rrrGGgrrrGGg"/>
        <phase duration="4" state="rrryygrrryyg"/>
        <phase duration="6" state="rrrrrGrrrrrG"/>
        <phase duration="4" state="rrrrryrrrrry"/>
        <phase duration="31" state="GGgrrrGGgrrr"/>
        <phase duration="4" state="yygrrryygrrr"/>
        <phase duration="6" state="rrGrrrrrGrrr"/>
        <phase duration="4" state="rryrrrrryrrr"/>
    </tlLogic>
    <tlLogic id="1/2" type="static" programID="0" offset="0">
        <phase duration="31" state="GgrrGG"/>
        <phase duration="4" state="ygrryy"/>
        <phase duration="6" state="rGrrrr"/>
        <phase duration="4" state="ryrrrr"/>
        <phase duration="31" state="rrGGGr"/>
        <phase duration="4" state="rryyyr"/>
    </tlLogic>
    <tlLogic id="2/1" type="static" programID="0" offset="0">
        <phase duration="31" state="GGGgrr"/>
        <phase duration="4" state="yyygrr"/>
        <phase duration="6" state="rrrGrr"/>
        <phase duration="4" state="rrryrr"/>
        <phase duration="31" state="GrrrGG"/>
        <phase duration="4" state="yrrryy"/>
    </tlLogic>
```

Appendix G

Retrieve controlled links for a given traffic light

In order to control traffic signal sequences, all the information related to controlled link such as incoming lane and outgoing lane is required. The relevant code segments and explanation on how to retrieve controlled link information for a given traffic light from SUMO via TRACI and TRASMAPI is given here.

TRASMAPI acts as a Java API to communicate with TRACI API in SUMO to get the traffic information of the traffic control simulation dynamically.

Once we connect to the SUMO simulation environment via a TCP connection using given port as client, we can communicate with SUMO using TCP messages. TRACI API provides a protocol, either to control the environment or to retrieve information using commands that are issued as TCP messages. TRACI command to retrieve traffic light information is 0xa2. For a given command, there will be set of variables that represents different details related to that particular command. Variable to get links controlled by a traffic light is 0x27. When, SUMO receives 0x27 command, it will return the incoming lane, outgoing lane and via lane for a given traffic light that are embedded as a CompoundObject as TCP message called 0xb2 response variable. The structure of the CompoundObject is given below.

integer	controlled links	...	controlled links
Length (number of signals)	**links controlled by signal 0**	**...**	**links controlled by signal n-1**

Controlled links:

int	stringlist	...	stringlist
number of controlled links	**link 0**	**...**	**link n-1**

Source:
http://www.sumo.dlr.de/wiki/TraCI/Traffic_Lights_Value_Retrieval#Command_0xa2:_Get_Traffic_Lights_Variable

Relevant code segments for the above custom implementation in TRASMAPI is given below.

```
// Receive controlled links information from SUMO
public ArrayList<ControlledLink> getControlledLinks(){

        Command command = new
        Command(Constants.CMD_GET_TL_VARIABLE);
        Content content = new Content(Constants.TL_CONTROLLED_LINKS,id);

        command.setContent(content);

        RequestMessage request = new RequestMessage();
        request.addCommand(command);

        ResponseMessage response = SumoCom.query(request);
        Content rspContent = response.validate( (byte)
        Constants.CMD_GET_TL_VARIABLE, (byte)
        Constants.RESPONSE_GET_TL_VARIABLE,
        (byte)  Constants.TL_CONTROLLED_LINKS, (byte)
        Constants.TYPE_COMPOUND);

        controlledLinks = rspContent.getControlledLinksFromCompoundObject();

        return controlledLinks;
}

// Retrieve controlled links information from CompoundObject in TRACI
public ArrayList<ControlledLink> getControlledLinksFromCompoundObject(){

        ArrayList<ControlledLink> controlledLinks = new
        ArrayList<ControlledLink>();

        int currentPointer = 5;
        int skipCount = 6;
        int integerCount = 4;

        int numberOfSignals = readInt(currentPointer);
        currentPointer += integerCount;

        for (int signal=0; signal < numberOfSignals; signal++)
        {
                ControlledLink link = new ControlledLink();

                currentPointer = currentPointer + skipCount + integerCount;
                String incoming = new String();
                int linkCharLength_incoming = readInt(currentPointer);
                currentPointer += integerCount;
```

```
                for (int character=0; character < linkCharLength_incoming;
                character++){
                        byte charact = varValue.get(currentPointer);
                        incoming += (char) charact;
                        currentPointer++;
                }
                link.incoming = incoming;
                String outgoing = new String();
                int linkCharLength_outgoing = readInt(currentPointer);
                currentPointer += integerCount;

                for (int character=0; character < linkCharLength_outgoing;
                character++){

                        byte charact = varValue.get(currentPointer);
                        outgoing += (char) charact;
                        currentPointer++;
                }
                link.outgoing = outgoing;

                String tls_index = new String();
                int linkCharLength_tls = readInt(currentPointer);
                currentPointer += integerCount;

                for (int character=0; character < linkCharLength_tls;
                character++){
                        byte charact = varValue.get(currentPointer);
                        tls_index += (char) charact;
                        currentPointer++;
                }
                link.tls_index = tls_index;
                controlledLinks.add(link);
        }
        return controlledLinks;
}
```