University of TRAFFIC: A highly configurable traffic operator model of the University of Florida

Andres Espinosa
Industrial and Systems Engineering
University of Florida
andresespinosa@ufl.edu

***Abstract*—**

I. Introduction

II. Implementation

A. Data Structures

In order to implement the traffic model, objects are split into four different categories:

1) Vehicles: Cars, mopeds, and buses are classified as vehicles and therefore share similar properties with each other. Each vehicle has the following: *max_speed, entrance, exit, paths* parameters. Cars are defined as the standard vehicle and are the parent object for mopeds and buses. Each car enters randomly though one of the *entrance* intersections and are assigned to an *exit* intersection. In order to determine the route they take, they choose which of their *paths* has the smallest number of cars on it.

In addition to the parameters defined in the `Car` object, mopeds and buses have slight differences that impact their behavior. Mopeds are different than the standard car as they take up less space. In order to simulate how mopeds tend to slowly go around pedestrians instead of fully waiting for them to be out of the crosswalk, mopeds can proceed through roundabouts and stop signs regardless of pedestrian activity. Buses are unique as they follow a generally rigid schedule. This means that each bus has only one *path* to select from and an additional *entrance_time*. Both buses and mopeds also have different roundabout behavior as mentioned in the next section

2) Intersections: Intersections are defined as any node where more than two sections of road meet. Roundabouts, stop signs, and traffic lights are defined as the possible intersection types. Each intersection behaves significantly differently and impacts traffic accordingly.

Roundabouts are full circles with an entrance and exit on each end that intersects a road. When a car enters a roundabout, it occupies 45 degrees of space on the circle and travels at 30 degrees per second until it has exited on its desired path. A car will enter a roundabout if it has a free 90 degree section to enter. A moped will enter a roundabout if it has a free 60 degree section to enter. A bus will enter a roundabout if it has a free 120 degree section to enter.

Stop signs are intersections where each car must slow down to 0 speed before entering. Cars exit stop signs in a FIFO (First-in-first-out) queue.

Traffic lights are a complex intersection type defined by how they choose which cars can flow freely past the node and which must stop. In order to model different traffic light behaviors, three sub-types are created:

- **Timer**: Parameterized by τ, the traffic light switches from green to red every τ seconds.
- **Sensor**: Parametrized by K, the traffic light switches to green if more than K cars are waiting at any given time.
- **Sensor-Timer**: Parametrized by τ, K, the traffic light switches to green if more than K cars are waiting at any given time or τ time has passed.

3) Roads: Roads are defined as the one-way arcs between different intersections. Each road has a length l.

4) Extra: Some extra factors that impact traffic that are implemented in this model are pedestrian crosswalks and parking lots.

Pedestrian crosswalks are entrances of an intersection that are randomly blocked. Cars and buses must wait for the pedestrians to exit the crosswalk before they must enter the intersection. Mopeds can still enter intersections if a pedestrian crosswalk is blocked but must slow down to 3 mph.

Parking lots are nodes along different arcs where there is a 5% chance that a car entering the node does not exit (the car parks).

III. Results

IV. Conclusion and Future Work

References

[1] A. Sheerazi, ”PLANTTRAITS2024 – A Kaggle competition,” Medium, [Online]. Available: https://medium.com/@adil.she/planttraits2024-a-kaggle-competition-8c6731003356

[2] Awsaf, AyushiSharma, HCL-Jevster, inversion, Martin Görner, Teja Kattenborn. (2024). PlantTraits2024 - FGVC11. Kaggle. https://kaggle.com/competitions/planttraits2024

[3] Wolf, S., Mahecha, M.D., Sabatini, F.M. et al. Citizen science plant observations encode global trait patterns. Nat Ecol Evol 6, 1850–1859 (2022). https://doi.org/10.1038/s41559-022-01904-x

[4] Moles, A.T., Xirocostas, Z.A. Statistical power from the people. Nat Ecol Evol 6, 1802–1803 (2022). https://doi.org/10.1038/s41559-022-01902-z

[5] Schiller, C., Schmidtlein, S., Boonman, C. et al. Deep learning and citizen science enable automated plant trait predictions from photographs. Sci Rep 11, 16395 (2021). https://doi.org/10.1038/s41598-021-95616-0

[6] Tan, M., Le, Q. V. (2021, June 23). EFFICIENTNETV2: Smaller models and faster training. arXiv.org. https://arxiv.org/abs/2104.00298

[7] He, Kaiming, et al. ”Identity mappings in deep residual networks.” Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands. https://arxiv.org/abs/1603.05027

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2012, doi: https://doi.org/10.1145/3065386.

[9] rui314, ”Kaggle Notebook Tabular Data Deep Learning” Kaggle. https://www.kaggle.com/code/rui314/planttraits-resnet18

[10] Michalinahulak, ”Tabular Data Deep Learning,” Kaggle. https://www.kaggle.com/code/michalinahulak/tabular-data-deep-learning/notebook

[11] Richolson, ”Tabular ImageNet Multi-Target Regression,” Kaggle. https://www.kaggle.com/code/richolson/tabular-imagenet-multi-target-regression/notebook

[12] Mark Wijkhuizen, ”PlantTraits2024 EDA and Training Pub,” Kaggle. https://www.kaggle.com/code/markwijkhuizen/planttraits2024-eda-training-pub

[13] Hdjojo, ”Modified PlantTraits2024 EDA and Training,” Kaggle. https://www.kaggle.com/code/hdjojo/modified-planttraits2024-eda-training/notebook

[14] Richolson, ”ImageNet Tabular XGBoost,” Kaggle. https://www.kaggle.com/code/richolson/imagenet-tabular-xgboost/notebook

[15] A. Howard et al., “Searching for MobileNetV3,” arXiv.org, 2019. https://arxiv.org/abs/1905.02244

[16] [3]K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016. Available: https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

[17] D. Reis, J. Kupec, J. Hong, and A. Daoudi, “Real-Time Flying Object Detection with YOLOv8.” Accessed: Apr. 29, 2024. [Online]. Available: https://arxiv.org/pdf/2305.09972

[18] C.-Y. Wang, H.-Y. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I-Hau. Yeh, “CSPNet: A New Backbone that can Enhance Learning Capability of CNN.” Available: https://openaccess.thecvf.com/content_CVPRW_2020/papers/w28/Wang_CSPNet_A_New_Backbone_That_Can_Enhance_Learning_Capability_of_CVPRW_2020_paper.pdf

[19] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System.” Accessed: Apr. 29, 2024. [Online]. Available: https://arxiv.org/pdf/1603.02754