Evaluation program

A demand generation function takes a demand scenario (which determines the temporal distribution of the demand) and demand level arguments as input and produces lists of demand. The i-th entry in the list is the OD demand departing in the i-th interval. Each OD pair can have its own time varying demand level and scenario in principle, but this has to be hard-coded into the module for now. For simplicity in this experiment, we assume that all OD pairs share the same level and scenario.

A disruption generation function takes a disruption scenario as input and outputs a list of probabilities. The i-th entry in the list is the probability of disruption ending in the i-th interval.

A mitigation plan generation module contains LLA, BB, BM and ITM algorithms. They take the initial state of the system as input and output the strategies for the whole horizon. The strategy is represented as lists of y's and x's, where $y[i]$ is a vector of the fleet size configurations at simulation time i if a disruption has not ended. We assume that once a disruption ends, all fleet sizes go back to initial settings – the normal level. x's have a similar interpretation.

The evaluation module takes time-dependent demand, disruption distribution, and mitigation plans as input, and outputs the user and operator cost. Time is discretized into one minute intervals when accumulating the user costs. The user demands in each one-minute interval are assigned according to the capacity constraints at that time. Not enough capacity on the shortest path means that users will detour to longer distance paths. User wait cost depends on average headway.

The main function is called by command *main.py [Q_scenario] [q_0] [q_max] [T_scenario] [alpha]*, where *Q_scenario* is the name of demand pattern. *q_0* and *q_max* are used to specify the level of demand. *T_scenario* is the name of the disruption pattern. The main function invokes network, demand, and disruption generation functions in sequence. Then the disruption mitigation strategy is generated at the beginning of the horizon. The program loops over time and evaluates user costs. Note that we assume the simpler case of demand being known and deterministic, hence the only stochastic factor is the disruption duration. Therefore, we can always generate the strategy for the whole horizon from the beginning. Under our assumptions, the returned strategy includes all the information that an operator needs to know to act when the uncertainty of disruption unfolds in the future. More complex models considering stochastic demand along with the corresponding simulation are possible and reserved for future work. The disruption ending time is simulated according to the distribution output by disruption generation function. See Figure 12 for the data flow diagram.

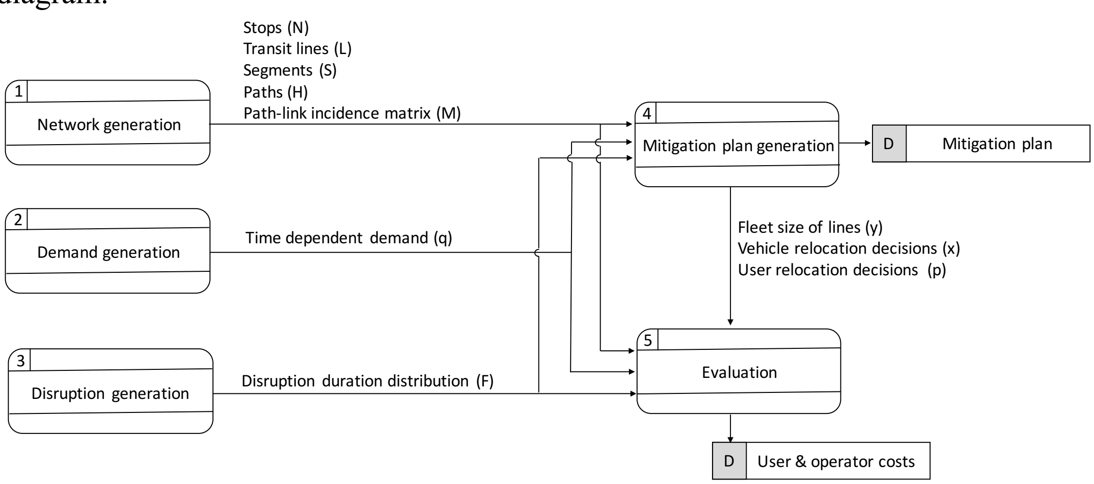

Figure 12. Data flow diagram of model evaluation