Second fORged-by-Machines Contest (2020)

Katherine Adams and Akhilesh Soni

Introduction

We model the given multi-commodity network flow optimization problem using a path based formulation. By commodity, we mean an origin-zip demand pair. We aggregate customer demand based on zip level. Using the given 365 datasets of customer demands, we approximate zip locations from the customers locations in that zip, weighting each customer location by its demand. We define a set of commodities K which is a set of all possible origin-zip pairs. Each commodity $k \in K$ is to be fulfilled by a set of candidate paths. We define path as a set of arcs connecting an origin node to a destination node directly or via transshipment node. Note that our paths don't include arcs between destination nodes and zip locations. We ensure through a set of constraints that if a path $p \in P$ fulfills a commodity $k \in K$, then destination of path p should have been mapped to zip destination of commodity k. We next define a set of arcs A which consist of arcs from origin ($o \in O$) $\rightarrow$ transshipment node ($t \in T$), transshipment node $\rightarrow$ destination node ($d \in D$), and origin $\rightarrow$ destination node.

Uncertain Data

A preliminary data analysis of demand for all commodities $k \in K$ or origin-zip pairs showed that there is a significant amount of uncertainty. We plotted histograms for commodity demands and did probability distribution fits for each commodity k (origin-zip) pair using normal and exponential distributions. We selected the best distribution and corresponding parameters based on goodness of fit tests (χ^2) for each commodity. We identified these probability distribution in order to randomly sample demands for each commodity while using a Sample Average Approximation algorithm which is discussed later. However, before that, we first present an extensive form (scenario based) model.

Extensive Formulation Model (EM)

We consider a finite and reasonable set of scenarios S. These scenarios are generated by randomly sampling the demand for each commodity from the fitted distributions. We assume that demands for different commodities are independent from each other.

Decision Variables

- $y_a \in Z^+$: Integer variable to keep track of number of trucks to run on arc $a \in A$
- $u_{zd} \in \{0, 1\}$: 1 if zip $z \in Z$ is served by destination node $d \in D$
- $r_k^s \in R^+$: Distance traveled to fulfill commodity $k \in K$ in scenario $s \in S$
- $x_{p,k}^s \in [0, 1]$: Fraction of commodity $k \in K$ fulfilled by path $p \in P$ in scenario $s \in S$
- $\text{unfulfilled}_k^s \in [0,1]$: Fraction of commodity $k \in K$ unfulfilled in scenario $s \in S$
- $l_{\max}^s$: Maximum number of packages fulfilled by a single destination node in scenario $s \in S$
- $l_{\min}^s$: Minimum number of packages fulfilled by a single destination node in scenario $s \in S$

Constraints

$\sum_{d \in D} u_{z,d} = 1$	$\forall z \in Z$	(zip destination assignment)
$r_k^s \geq p_{\text{length}} \cdot x_{p,k}^s + \text{distance}(k_{\text{zip}}, d) \cdot u_{k_{\text{zip}},d}$	$\forall k \in K, d \in D, p \in P(k, d)$	(distance travelled)
$\sum_{p \in P(k)} x_{p,k}^s + \text{unfulfilled}_k^s = 1$	$\forall k \in K, s \in S$	(commodity fulfilled)
$x_{p,k}^s \leq y_{a_p}$	$\forall p \in P(k), k \in K, s \in S$	(path exists if arc open)
$\sum_{k \in K} \sum_{p \in P(k,a)} k_{\text{quantity}}^s x_{p,k}^s \leq 1000 y_a$	$\forall a \in A, s \in S$	(truck capacity)
$u_{z,d} \geq x_{p,k(p)}^s$	$\forall p \in P(d, z), z \in Z, d \in D$	(path destination mapped to zip)
$l_{min}^s \leq \sum_{k \in K} k_{\text{quantity}}^s u_{k_{\text{dest}} d}$	$\forall d \in D, s \in S$	(min load served by a warehouse)
$l_{max}^s \geq \sum_{k \in K} k_{\text{quantity}}^s u_{k_{\text{dest}} d}$	$\forall d \in D, s \in S$	(max load served by a warehouse)

Objective In our model, we combined cost per delivered package and cost of missed packages into the first objective below. The second objective listed is the distance traveled by delivered packages. The third objective seeks to minimize the maximum imbalance between warehouse loads. Expressions for all three objectives are given below:

$$\text{obj cost:} \quad \sum_{a \in A}(100 + 2d_a)y_a + \frac{1}{|S|}\sum_{s \in S}\sum_{k \in K} 1000 * q_k^s \text{ unfulfilled}_k^s$$

$$\text{obj distance:} \quad \frac{1}{|S|}\sum_{k \in k, s \in S} r_k^s$$

$$\text{obj load:} \quad \frac{1}{|S|}\sum_{s \in S}(l_{max}^s - l_{min}^s)$$

We gauged the best values for each objective by solving them separately. We then set bounds for each one and solved the model sequentially for each objective using Gurobi's function setObjectiveN.

Sample Average Approximation

We fitted distribution curves to our commodity demands. As each origin-zip pair can have any demand from the fitted distribution, there are exponentially many scenarios possible. Thus, we use sample average approximation (SAA) approach which is describe in Algorithm 1.

Algorithm 1: SAA for network design

Input: Number of batches (M), Number of scenarios ($|S|$)
Output: Optimal solution for first stage decisions ($y_a, u_{z,d}$)
for $i = 1 : M$ **do**
  generate S scenario ($\xi_1, \xi_2, \ldots, \xi_S$), demand is sampled from the fitted distributions;
  `/* Note` ξ_i `is a vector with demand realization for each origin-zip pair */`
  solve extensive form model, EM($\xi_1, \xi_2, \ldots, \xi_S$) ;
  store candidate solution ($y_a^i, u_{z,d}^i$) and objective values O^i
end
`/* select best candidate solution */`
$y_a^*, u_{z,d}^* = \arg\min\{\mathrm{O}^i(y_a^i, u_{z,d}^i) : i \in \{1, 2, \ldots, M\}\}$

Note that we sampled separately to construct a Confidence Interval on our best candidate solution. CI was constructed after fixing the first stage solution and randomly generating new demand scenarios and solving EM with first stage decisions fixed. This is compared against the EM solution for this new sampled demand but without first stage decisions fixed.

For our experiments, we used S in the range of 10-20 and M in the range of 5-10. We kept a time limit of 10000s for each batch run. We conducted experiments on a 16 GB, 2.8 GHz Quad-Core i-7 machine.

Heuristics

To improve computational efficiency, we use the following heuristics:

- Warm-start: Due to a large number of binary variables and scenarios, it is challenging to solve the extensive form of our model multiple times. To reduce solution time and improve the quality of solutions when time limit is reached, we used warm-start with the optimal solution obtained from solving a deterministic model with mean demands for each origin-zip pair (based on fitted probability distribution).
- Practical constraints: We add the following constraints in our model to eliminate some paths
  - A zip should be served by one of the four closest destination nodes
  - Forcing unfulfilled variables to be 0. Cost of missing a package is much higher compared to running a truck for a single package, hence we forced the model to not let any fraction of the commodity be unfulfilled.