15 Pizza guy

The best pizzeria in town delivers pizzas at home. Received orders specify the number of pizzas to be delivered, the address of delivery and the desired time of delivery. Assume that we know k addresses, extracted from the local map and encoded as $n_1, n_2, n_3, \ldots, n_k$ nodes. Times of delivery have 15 minutes granularity (from 19.00 to 22.00) and the delivery window is up to 30 minutes later than delivery time.

Addresses are connected by edges that contain travel time between nodes. Assume that the pizzeria is n_1.

Assume that g deliverers can carry at most 16 pizzas in their bags.

The problem is to find the optimal schedule for deliverers so that they can deliver all pizzas and the total travelled distance is minimal. A deliverer can perform multiple travels from and to the pizzeria, even within the delivery window.

1. Write a Minizinc program that solves the problem.

2. Prepare your benchmark suite as follows. Generate randomly 5 towns of increasing size. Edges can't be totally random since they represent a town. In a planar representation of the graph, you can have some edge crossing (e.g., there is a bridge) but not too many. Moreover, edge travel time should satisfy topology constraints (triangular inequality). For each town generate 10 possible input sets. Choose k, g in such a way that at least two travels are needed for each deliverer.

3. Run your Minizinc encoding on all the instances. Use a timeout of 5 minutes for each test ("configuration" option in Minizinc)

4. Choose one configuration that runs in a couple of minutes and try alternative search primitives on it. Use the most promising one for more difficult instances and report what happens.

5. Write a 6–10 pages report containing your models (and the reasons for some choices) and a presentation of the execution results. Prepare the programs and the benchmark instances used in a unique zip file