Google OR-Tools

https://google.github.io/or-tools/

https://github.com/koftezz/google-ortools-ex

https://how-to.aimms.com/Articles/332/332-Miller-Tucker-Zemlin-formulation.html

https://arxiv.org/pdf/1606.01935.pdf

OR-Tools is an open source software for optimization problems in vehicle routing, flows, integer and linear programming, and constraint programming. We can call commercial solvers such as Gurobi or CPLEX, or open-source solvers such as SCIP, GLPK, or Google's GLOP and award-winning CP-SAT.

There are 5 modules.

- Algorithms module
- Graph module
- Linear Solver module
- CP-SAT module
- Routing module
- Domain module

This is a Two-index vehicle flow formulation with Miller-Tucker-Zemlin subtour elimination constraint.

$$\min \quad \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_{ij} \tag{2.1}$$

$$\text{s.t.} \quad \sum_{\substack{j=1 \\ j \neq i}}^{n+1} x_{ij} = 1, \qquad i = 1, \ldots, n, \tag{2.2}$$

$$\sum_{\substack{i=0 \\ i \neq h}}^{n} x_{ih} - \sum_{\substack{j=1 \\ j \neq h}}^{n+1} x_{hj} = 0, \qquad h = 1, \ldots, n, \tag{2.3}$$

$$\sum_{j=1}^{n} x_{0j} \leq K, \tag{2.4}$$

$$y_j \geq y_i + q_j x_{ij} - Q(1 - x_{ij}), \qquad i, j = 0, \ldots, n+1, \tag{2.5}$$

$$q_i \leq y_i \leq Q, \qquad i = 0, \ldots, n+1, \tag{2.6}$$

$$x_{ij} \in \{0, 1\}, \qquad i, j = 0, \ldots, n+1. \tag{2.7}$$

$c_{i,j}$: cost matrix

q_i: demand

$x_{i,j}$: binary decision variable $x_{i,j}$ that assumes value 1 if and only if there is a route that goes from customer i to j directly.

y_i: continuous decision variable corresponding to the cumulated demand on the route that visits ode

- Consider a set of customers represented by C = {1, . . . , n}, such that a positive demand
  is associated to each customer i ∈ C.
- To service these customers, we have to design routes for a fleet with K vehicles available in a single depot.
- Each route must start at the depot, visit a subset of customers and then return to the depot.
- All customers must be visited exactly once.
- Each vehicle has a maximum capacity Q, which limits the number of customers it can
  visit before returning to the depot.
- We assume a homogeneous fleet of vehicles.

- Define set of nodes associated to customers in C and to the depot nodes 0 and n + 1. We use two nodes to represent the same single depot and impose that all routes must start on 0 and return to n + 1.
- The cost of crossing an arc (i, j) $\in$ E is denoted by cij . Each node has a demand qi, such that qi > 0 for each i $\in$ C and q0 = qn+1 = 0.
- The objective of the problem is to determine a set of minimal cost routes that satisfies all the requirements defined above.
- Constraints (2.2) ensure that all customers are visited exactly once.
- Constraints (2.3) guarantee the correct flow of vehicles through the arcs, by stating that if a vehicle arrives to a node h $\in$ N , then it must depart from this node.
- Constraint (2.4) limits the maximum number of routes to K, the number of vehicles.
- Constraints (2.5) and (2.6) ensure together that the vehicle capacity is not exceeded.
- The objective function is defined by (2.1) and imposes that the total travel cost of the routes is minimized.
- Constraints (2.5) also avoid subtours in the solution, i.e. cycling routes that do not pass
  through the depot. Different types of constraints are proposed in the literature to impose
  vehicle capacities and/or avoid subtours [17]. The advantage of using (2.5) and (2.6) is that
  the model has a polynomial number of constraints in terms of the number of customers.
  However, the lower bound provided by the linear relaxation of this model is known to be
  weak in relation to other models