A data-structure covering a large scope of the taxonomy for routing applications

Atoptima

November 6, 2018

Routing applications in logistics come in a rich panel of variations. Where possible, we consider the most general case that can be accommodated in our generic data structure; for otherwise, we make restrictive assumptions on the problem variant.

So, we make the following <u>restrictions</u>, we consider:

- a **static and deterministic** problem as opposed to stochastic or with dynamically incoming data a multi-period horizon;

- **pure routing** decisions as opposed to integration with driver timetabling, inventory-management, production scheduling; in particular the framework does not implement restrictions due to driver regulation or packing placement issues, such as a 3D positioning in the vehicle (such issue are not trivial); in both suc case only surrogate relaxation of such constraint can be incorporated;

- a **single route** within the time horizon (the extension to the multiple route case is not straightforward).

But our framework includes the following <u>extentions</u> of the basic problem:

- **multiple depots** (single depot being a special case);

- **heterogenous vehicle fleet** (a homogenous fleet being a special case);

- **bounded number of vehicles** (the unbounded case being a special case where the bounds are large enough to induce no restrictions on the problem);

- **time lag on shipments:** this is modeled by a max duration for a shipment as in dial a ride;

- **time-windows:** can be defined on pickups, deliveries, vehicle availability (f.i. for driver working day);

- **sequencing:** our framework recognize the backhauls special case; as well as prohibited sequence;

- **conflicts:** our framework recognize forbidding products to share a route;

- **split-quantity-option:** it is on if requests can be fulfilled by different routes;

- **request-cover-mode:** a request can be mandatory, or optional with a price reward for covering it;

- **mixing single-commodity and multi-commodity mode:** In the single commodity, the goods that are transported are undistinguishable. So a delivery can be done from any pickup point. When the requests are only of one type: either all are pickups or all are deliveries; then the problem should be understood as a single commodity problem, are there is no need to distinguish the goods. A multi-commodity model is required in the presence of "Pickup and Delivery" requests. The latter is associated with a specific pickup point and to a specific delivery point. When some goods are replaceable by others, then the associated delivery (resp. pickup) can have a choice of pickup (resp. delivery) points. So a complex request may be defined that involves several alternative pickup-point and/or several alternative delivery points. This complexity can make specific sense along with the split quantity option. In particular, in the multi depot case, goods can typically be delivered from (resp. delivered to) several of them. Our shipment-request model allows us to mix single-commodity and multi-commodity types.

- **service-times:** we distinguish access time and pickup or delivery points, setup time and set-down time, the sum making up the service time.

- **multiple compartments** in the vehicle (single compartment being a special case), with loading mode option. For instance, one can model a restriction of one request per compartment for liquid transportation; then, the request can be accommodated only if the compartment capacity is larger than the request capacity requirement. Or a vehicle can have a trailer; then both the vehicle and the trailer have a maximum weight capacity, while each request has its own weight.