Route Optimization Model
Pathfinder
December 7, 2015
Version 1.1

Pathfinder models route optimization as a linear programming problem. The function to be optimized and the constraints will be specific to a specific cluster for an application.

Parameters

Pathfinder considers a route as a chain of "route actions" where a route action is either a vehicle starting position, a commodity pickup or a commodity drop off. A route action corresponds to exactly one location. One commodity transport request generates exactly two route actions each with a location: pickup and dropoff.

Pathfinder distinguishes between two classes of properties of vehicles and commodities. Capacity parameters enforce limits on the commodities that can be simultaneously transported by a vehicle. Routing parameters are used in custom optimization functions.

- $c_{i,j}$ - The capacity of vehicle *i* for capacity parameter *j*.
- $v_{k,j}$ - The value change of capacity parameter *j* incurred by performing route action *k*. Note that for dropoff route actions, this will be negative.
- $vp_{i,m}$ - The value of routing parameter *m* for vehicle *i*. These are properties of the vehicle that might affect the optimization function, such as gas mileage.
- $cp_{k,m}$- The value of routing parameter *m* for commodity *k*. These are properties of the route action that might affect the optimization function, such as "passenger priority".
- d_{k_1,k_2} - The driving distance from route action k_1 to route action k_2.
- t_{k_1,k_2} - The driving time from route action k_1 to route action k_2.

Variables

Note that these variables
- $x_{k_1,k_2,i}$ – 1 if the route for vehicle *i* contains both k_1 and k_2 and they are in consecutive order, else 0.
- $y_{k_1,k_2,i}$ – 1 if the route for vehicle *i* contains both k_1 and *k2* and k_1 occurs before k_2, else 0.

Problem

Let *S* denote the set of routing actions that are vehicle starting locations. Let *T* denote the set of all other routing actions.

$$\min f(d_{k_a,k_b}, t_{k_a,k_b}, vp_{i,m}, cp_{k,m})\,, s.t.$$
$$\sum_{k_1} v_{k_1,j} y_{k_1,k_2,i} \leq c_{i,j}, \forall i, j, k_2$$

$$\sum_{i,k_1} x_{k_1,k_2,i} = 1, \forall k_2 \in \mathrm{T}$$

$$Mx_{k_1,k_2,i} > \left(\sum_{k_3} y_{k_2,k_3,i} - y_{k_1,k_3,i}\right) - 1, \forall k_1, k_2, i$$

$$x_{k_1,k_2,i} = 0, \forall i, k_2 \in S$$

$$y_{k_1,k_2,i} = 0, \forall i, k_2 \in S$$

$$x_{k_1,k_2,i} \geq 0, \forall i, k_2 \in T$$

$$x_{k_1,k_2,i} \leq 1, \forall i, k_2 \in T$$

$$y_{k_1,k_2,i} \geq 0, \forall i, k_2 \in T$$

$$y_{k_1,k_2,i} \leq 1, \forall i, k_2 \in T$$

The first set of constraints enforces all that all vehicle capacities are respected for every route segment. The second set of constraints ensures that every commodity is picked up and dropped off exactly once. The third constraint enforces consistency between *x* and *y*. The remaining equations ensure that *x* and *y* are binary variables and that start actions do not occur in the middle of routes.

Objective function

The difficult part of the problem is determining a suitable function *f*. One example would be minimizing total distance travelled on all routes:

$$\min \sum_{k_1,k_2,i} d_{k_1,k_2} x_{k_1,k_2,i}$$

Solution

A Julia service solves the optimization problem using the GNU Linear Programming Kit (GLPK, https://gplkjl.readthedocs.org/en/latest/glpk.html). We will use the HttpServer.jl package to expose the routing engine as a web service (https://github.com/JuliaWeb/HttpServer.jl).

An example of a Julia web service that solves a linear programming problem can be seen at https://github.com/aj-michael/sudoku.jl.