IEOR 151 – Lecture 18
Savings Algorithm

1 Problem Formulation

Recall our formulation of the vehicle routing problem: There are a set of depots, vehicles, and delivery locations, and the problem is to optimally design routes for the vehicles from the depots to delivery locations. This is represented by an undirected graph $G = (V, E)$ with vertices representing locations and edges representing paths between locations. *One important facet of this model is that each vertex is connected by an edge.* The depot is located at vertex v_0, and there are delivery locations at the remaining vertices. Furthermore, the edges e_{ij} are weighted by d_{ij}, which represents the distance between vertices v_i and v_j. The maximum number of vehicles is n, which would be the situation in which exactly one vehicle is assigned to each delivery location. Each vehicle has a maximum capacity of goods G, and each vehicle has a maximum distance D that it can travel. Finally, each delivery location v_i for $i \geq 1$ has a demand value w_i.

The vehicle routing problem is to find least-cost vehicle routes so that

1. Each delivery location is visited exactly once by exactly one vehicle;
2. All vehicles start and end at the depot;
3. Side constraints on maximum vehicle capacity and travel distance are satisfied;

2 Clarke and Wright's Savings Algorithm

Solving this problem exactly can be difficult, and so a number of heuristics have been developed. One of the conceptually simplest heuristics is Clarke and Wright's Savings Algorithm. The algorithm proceeds as following:

1. Make n routes: $v_0 \to v_i \to v_0$, for each $i \geq 1$;
2. Compute the savings for merging delivery locations i and j, which is given by $s_{ij} = d_{i0} + d_{0j} - d_{ij}$, for all $i, j \geq 1$ and $i \neq j$;
3. Sort the savings in descending order;

4. Starting at the top of the (remaining) list of savings, merge the two routes associated with the largest (remaining) savings, provided that:
    (a) The two delivery locations are not already on the same route;
    (b) Neither delivery location is interior to its route, meaning that both notes are still directly connected to the depot on their respective routes;
    (c) The demand G and distance constraints D are not violated by the merged route.
5. Repeat step (3) until no additional savings can be achieved.

3 Example

Consider the nodes described below, and note that the depot is located at node 0. Suppose we would like to solve this vehicle routing problem (VRP) using the savings algorithm, for the constraint that each vehicle has a capacity of 30 units (meaning it can carry less than or equal to 30 units).

Distance	Node 0	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Demand
Node 0									0
Node 1	4								12
Node 2	4	5.66							12
Node 3	2.83	6.32	2.83						6
Node 4	4	8	5.66	2.83					16
Node 5	5	8.54	8.06	5.39	3				15
Node 6	2	4.47	6	4.47	4.47	4.12			10
Node 7	4.24	3.16	7.62	7.07	7.62	7	3.16		8

We begin by computing the savings:

Node i	Node j	Savings
1	2	4 + 4 - 5.66 = 2.34
1	3	4 + 2.83 - 6.32 = 0.51
1	4	4 + 4 - 8 = 0
1	5	4 + 5 - 8.54 = 0.46
1	6	4 + 2 - 4.47 = 1.53
1	7	4 + 4.24 - 3.16 = 5.08
2	3	4 + 2.83 - 2.83 = 4
2	4	4 + 4 - 5.66 = 2.34
2	5	4 + 5 - 8.06 = 0.94
2	6	4 + 2 - 6 = 0
2	7	4 + 4.24 - 3.16 = 5.08
3	4	2.83 + 4 - 2.83 = 4
3	5	2.83 + 5 - 5.39 = 2.44
3	6	2.83 + 2 - 4.47 = 0.36
3	7	2.83 + 4.24 - 7.07 = 0
4	5	4 + 5 - 3 = 6
4	6	4 + 2 - 4.47 = 1.53
4	7	4 + 4.24 - 7.62 = 0.62
5	6	5 + 2 - 4.12 = 2.88
5	7	5 + 4.24 - 7 = 2.24
6	7	2 + 4.24 - 3.16 = 3.08

Next, we merge routes using the savings sorted in descending order and taking into account the capacity constraints of the vehicles gives:

Route	Demand	Length
~~0 – 1 – 0~~	~~12~~	
~~0 – 2 – 0~~	~~12~~	
~~0 – 3 – 0~~	~~6~~	
0 – 4 – 0	16	4 + 4 = 8
0 – 5 – 0	15	5 + 5 = 10
~~0 – 6 – 0~~	~~10~~	
~~0 – 7 – 0~~	~~8~~	
~~0 – 1 – 7 – 0~~	~~20~~	
0 – 2 – 3 – 0	18	4 + 2.83 + 2.83 = 9.66
0 – 1 – 7 – 6 – 0	18	4 + 3.16 + 3.16 + 2 = 12.32

The nodes visited on each route, the demand associated with that route, and the route length are given in the table above; note that routes in the solution are those that have not been crossed out. Also note that depending on how duplicate values are sorted, it is possible to get the routes $\{0--1--7--6--0, 0--3--4--0, 0--2--5--0\}$ as a solution which have lengths of 12.32, 9.66, 17.06, respectively.

4 More Information and References

The material in the first section of these notes follows that of the journal paper G. Laporte, "The Vehicle Routing Problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, pp. 234–358, 1992. The material in the last section of these notes follows the course textbook "Service Systems" by Mark Daskin.