Savings Algorithm

The Clarke and Wright savings algorithm is one of the most known heuristic for VRP. It was developed on [Clarke and Wright 1964] and it applies to problems for which the number of vehicles is not fixed (it is a decision variable), and it works equally well for both directed and undirected problems. When two routes ${(0,\ldots,i,0)}$ and ${(0,j,\ldots,0)}$ can feasibly be merged into a single route ${(0,\ldots,i,j,\ldots,0)}$, a distance saving ${s_{ij}=c_{i0}+c_{0j}-c_{ij}}$ is generated. The algorithm works at follows (the first step is equal in both parallel and sequential versions):

Step 1. Savings computation

- Compute the savings ${s_{ij}=c_{i0}+c_{0j}-c_{ij}}$ for ${i,j=1,\ldots,n}$ and ${i \neq j}$.
- Create ${n}$ vehicle routes ${(0,i,0)}$ for ${i=1,\ldots,n}$.
- Order the savings in a non increasing fashion.

Step 2. Best feasible merge (Parallel version)

Starting from the top of the savings list, execute the following:

- Given a saving ${s_{ij}}$, determine whether there exist two routes that can feasibility be merged:
  - One starting with ${(0,j)}$
  - One ending with ${(i,0)}$
- Combine these two routes by deleting ${(0,j)}$ and ${(i,0)}$ and introducing ${(i,j)}$.

Step 2. Route Extension (Sequential version)

- Consider in turn each route ${(0,i,\ldots,j,0)}$.
- Determine the first saving ${s_{ki}}$ or ${s_{jl}}$ that can feasibly be used to merge the current route with another route ending with ${(k,0)}$ or starting with ${(0,l)}$.
- Implement the merge and repeat this operation to the current route.
- If not feasible merge exists, consider the next route and reapply the same operations.
- Stop when not route merge is feasible.