# Transport voor een supermarkt keten

Simpel model, alle filialen met 1 vrachtwagen achter elkaar

* The Salesman problem (TSP) => ILP
  + Oplossen met de opslag locatie en alle filialen
    - Voor filialen in dezelfde plaats een reistijd van 5 min invoeren
      * Miller-Tucker-Zemlin formulation
      * <https://wikimedia.org/api/rest_v1/media/math/render/svg/f5956f5c05662044756c58c62cf6be5ebfd6c8a5>
        + Waar xij = 1 als er een weg is van stad i naar j, en 0 anders
        + n is het aantal steden
* Complicatie: Niet alle tomaten passen in 1 vrachtwagen
  + Hebben maar beperkt aantal vrachtwagens
* The Salesman problem moet worden opgedeeld in meerdere subproblemen

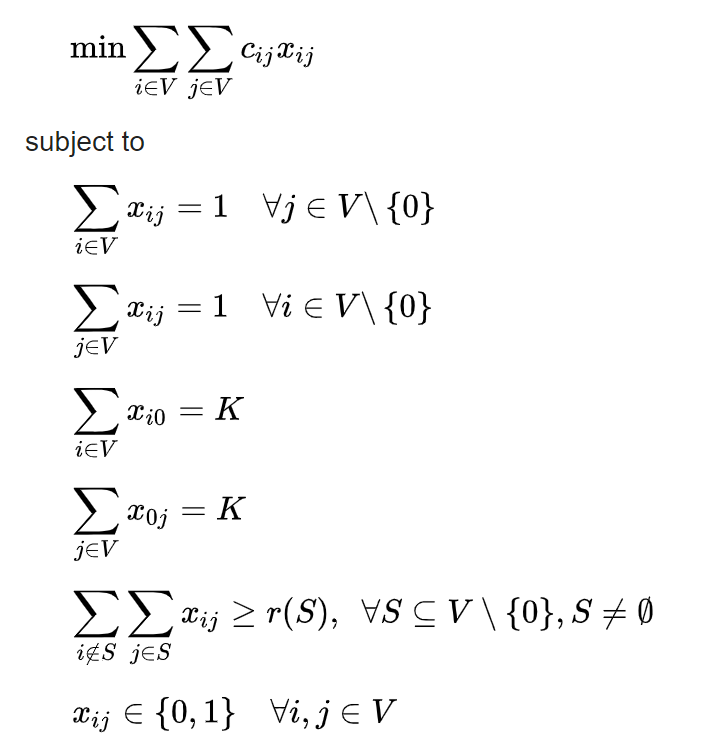
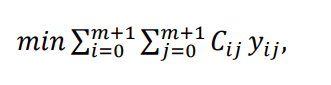
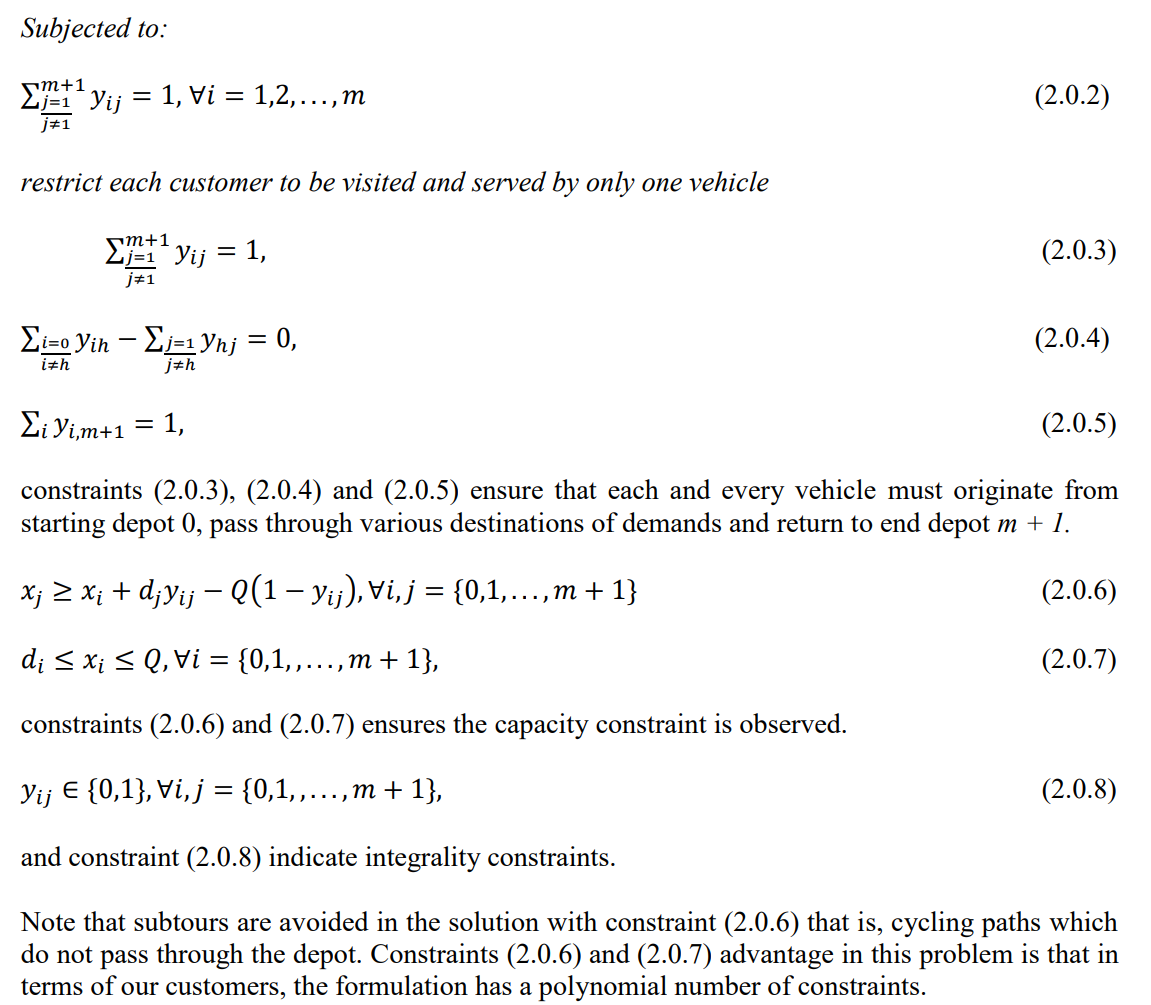
Uitgebreider model waar het netwerk opgedeeld word over verschillende vrachtwagens

* Multiple Traveling Salesman Problem (mTSP) => Mixed Integer Linear Programming
* Consider a graph G = (V,A), where V is the set of n nodes, and A is the set of edges. Associated with each edge (i,j) ∈ A is a cost (or distance) cij. We assume that the depot is node 1 and there are m salesmen at the depot. We define a binary variable xij for each edge (i,j) ∈A;  xij takes the value 1 if edge (i,j) is included in a tour and xij takes the value 0 otherwise. For the subtour elimination constraints, we define an integer variable ui to denote the position of node i in a tour, and we define a value p to be the maximum number of nodes that can be visited by any salesman.
  + Objective
    - Minimize ∑(i,j)∈A cijxij
  + Constraints
    - Ensure that exactly mm salesmen depart from node 1  
      ∑j∈V:(1,j)∈A x1j = m
    - Ensure that exactly mm salesmen return to node 1  
      ∑j∈V:(j,1)∈A xj1 = m
    - Ensure that exactly one tour enters each node  
      ∑i∈V:(i,j)∈A xij = 1,∀j∈V
    - Ensure that exactly one tour exits each node  
      ∑j∈V:(i,j)∈A xij = 1,∀i∈V
    - Include subtour elimination constraints (Miller-Tucker-Zemlin)  
      ui − uj + p ⋅ xij ≤ p − 1, ∀ 2≤i≠j≤n
* Dit verdeeld het probleem over een gelijk aantal steden per vrachtwagen, dus is nog niet 100% efficient
* Zet een maximum nummer vrachtwagens m

Laatste probleem is dat je een maxinum aantal nodes moet stellen per vrachtwagen. Dus moet je een soort gewichten systeem instellen aan de hand van de vraag aan tomaten per winkel.

* Elke winkel krijgt een gewicht met de vraag van de tomaten
* Constaint dat de som van de gewichten van de winkels die een vrachtwagen af gaat, niet groter is dan de maximale capaciteit van een vrachtwagen

Uitgebreider model: (capacitated)Vehicle Routing Problem (C)VRP

* Hier heeft een vrachtwagen geen onbeperkte capaciteit
*  => VRP
* 
* 
* <https://developers.google.com/optimization/routing/cvrp>

Bronnen :

<https://en.wikipedia.org/wiki/Travelling_salesman_problem>

<https://neos-guide.org/content/multiple-traveling-salesman-problem-mtsp>

<https://books.google.nl/books?id=k_iQyE3NME4C&pg=PA242&lpg=PA242&dq=mtsp+different+capacity+salesman&source=bl&ots=3jgIFq_mpf&sig=ACfU3U0g1pXBZAqw69V6mV3Mem2RGm7mOg&hl=nl&sa=X&ved=2ahUKEwjI9IXx3vTmAhUNCewKHdrMDcsQ6AEwA3oECAgQAQ#v=onepage&q=mtsp%20different%20capacity%20salesman&f=false>

<https://en.wikipedia.org/wiki/Vehicle_routing_problem>

<https://developers.google.com/optimization/routing/cvrp> => Met python code

<https://www.researchgate.net/publication/332370693_CAPACITATED_VEHICLE_ROUTING_PROBLEM>

<https://www.youtube.com/watch?v=7_-Xuq2xKdc>