Capacitated Vehicle Routing Problem with Pickup and Delivery

- Map
- Products
- Warehouses
- Orders
- Drones
- Commands
  - Load
  - Deliver

100 100 3 50 500	100 rows, 100 columns, 3 drones, 50 turns, max payload is 500u
3	There are 3 different product types.
100 5 450	The product types weigh: 100u, 5u, 450u.
2	There are 2 warehouses.
0 0	First warehouse is located at [0, 0].
5 1 0	It stores 5 items of product 0 and 1 of product 1.
5 5	Second warehouse is located at [5, 5].
0 10 2	It stores 10 items of product 1 and 2 items of product 2.
3	There are 3 orders.
1 1	First order to be delivered to [1, 1].
2	First order contains 2 items.
2 0	Items of product types: 2, 0.
3 3	Second order to be delivered to [3, 3].
1	Second order contains 1 item.
0	Items of product types: 0
5 6	Third order to be delivered to [5, 6]
1	Third order contains 1 item.
2	Items of product types: 2.

Example input file.

9	9 commands in total.
0 L 0 0 1	Drone 0: **load** one product 0 at warehouse 0.
0 L 0 1 1	Drone 0: **load** one product 1 at warehouse 0.
0 D 0 0 1	Drone 0: fly to customer 0 and **deliver** one product 0.
0 L 1 2 1	Drone 0: fly to warehouse 1 and **load** one product 2.
0 D 0 2 1	Drone 0: fly to customer 0 and **deliver** one product 2.
1 L 1 2 1	Drone 1: fly to warehouse 1 and **load** one product 2.
1 D 2 2 1	Drone 1: fly to customer 2 and **deliver** one product 2.
1 L 0 0 1	Drone 1: fly to warehouse 0 and **load** one product 0.
1 D 1 0 1	Drone 1: fly to customer 1 and **deliver** one product 0.

Example submission file.

Related Work

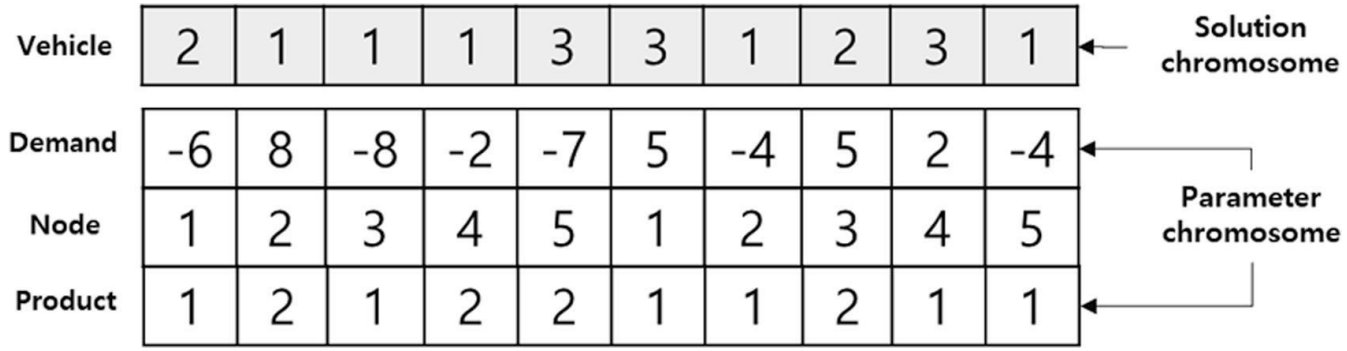
- 2020's Kaggle approach of same problem, created by Google: https://www.kaggle.com/c/hashcode-drone-delivery
- First Place solution, using Scala: https://www.kaggle.com/lyxthe/1st-place-solution | https://github.com/miraan/google-hash-code-qualification-round
- Second Place Solution, using Python: https://www.kaggle.com/royceda/drone-linear-prog-and-routing-heuristic-90-done
- Another Kaggle solution using Python: https://www.kaggle.com/spacelx/2020-hc-dd-2nd-place-solution-w-or-tools

- Google OR-Tools for VRP e CVRP: https://developers.google.com/optimization/routing/vrp; https://developers.google.com/optimization/routing/cvrp
- Google OR-Tools for CVRPPD: https://developers.google.com/optimization/routing/pickup_delivery
- LocalSolver : https://www.localsolver.com/benchmarkcvrp.html | https://www.localsolver.com/docs/last/exampletour/vrp.html

- CVRPPAD Hybrid Approach - https://link.springer.com/article/10.1007/s10479-017-2722-x | https://link.springer.com/content/pdf/10.1007/s10479-017-2722-x.pdf

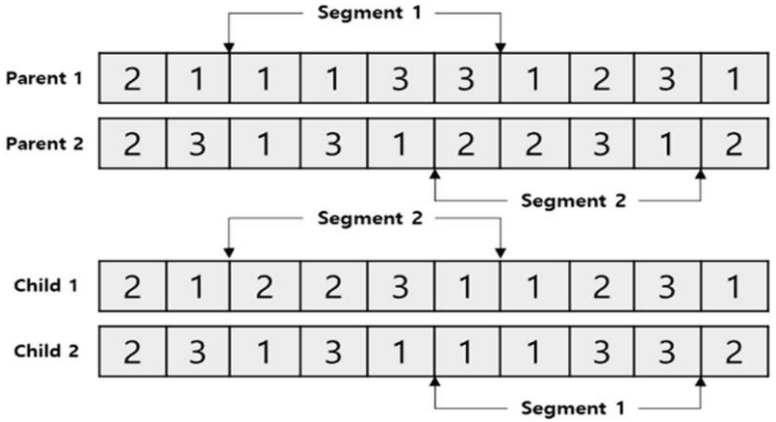
- Genetic Algorithm :
- A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries (2011): https://www.sciencedirect.com/science/article/pii/S0360835211003482
- A vehicle routing problem solved by using a hybrid genetic algorithm (2007): https://www.sciencedirect.com/science/article/pii/S0360835207001222
- Waiting strategy for the vehicle routing problem with simultaneous pickup and delivery using genetic algorithm (2020): https://www.sciencedirect.com/science/article/pii/S0957417420307429
- A hybrid genetic algorithm for the multi-depot vehicle routing problem (2005): https://www.sciencedirect.com/science/article/pii/S0952197607000887

Solution Representation



- Vehicle – If value is 0, no drone is assigned to that gene, but it can be used for future generations. If the value is greater than 0 and a integer, then a drone is assigned to the task represented by the gene
- Demand – weight of the product to be delivered, if the value is negative, or picked up, if positive
- Node – Identifier of a WH or a order. The order is identified and not the delivery point because multiple orders can have the same delivery point
- Product – Identifier of the product

Crossover functions



Cause changes to the drones trajectories

- A drone that pick ups product can deliver it to a different client if it is more beneficial.

Neighborhood/Mutation

Pq -> Product quantity

Dc -> Drone capacity

Cromossome Mutations:

- **Alter trajectories** by switching drones of 2 genes.
- **Unbalance the quantities** of 2 alleles of the same WH with the same item.
- **Add Supply genes**: If there are items in WH's that are not beeing picked up, adds a gene with a random quantity of that item, between [1, min(Pq,Dc)]
- **Join 2 genes:** They can be of type deliver or supply, but need to have the same node. After joining, the resulting gene can stay on the position of the first gene or the second.

Gene Mutations:

- **Split a gene** of type supplier or deliver into two genes, of either balanced or unbalanced quantities
- **Alter Associated Drone** by replacing the current drone identifier with another or by 0

Rigid Constraints

- Drone Route cannot surpass the defined number of turns
- Drone cannot have carry more weight than it’s capacity and cannot carry negative weight
- Drone cannot take from a WH more product that available
- Drone cannot pickup products from customer nodes, only from WH’s
- Drone can only deliver products that he is currently carriyng
- No order should receive more that than the intended products

Implementation work carried out

- Programming Language : Python
- Development Environment : JetBrains IntelliJ
- Already Implemented:
  - Parsing of input file
  - Creation of data structures to hold information