Selected Genetic Algorithms for Vehicle Routing Problem Solving

Group 6

108062226 蘇志翔

Table of Contents

- Paper introduction
- VRP(Vehicle Routing Problem) Introduction
- GA and its modifications for the VRP
- Conclusion

Paper information

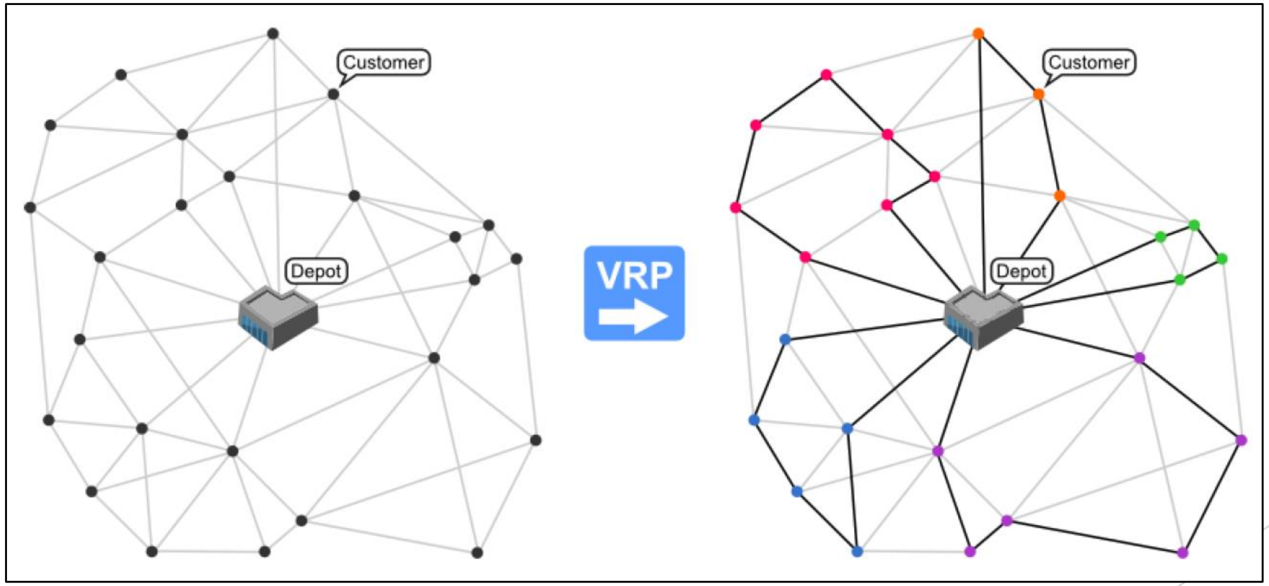
Information

- Author: J. Ochelska-Mierzejewska, A. Poniszewska-Mara ´nda, W. Mara´nda
- Journal: *Electronics 2021, MDPI*
- Published date: December 17, 2021
- Links: https://www.mdpi.com/2079-9292/10/24/3147

VRP(Vehicle Routing Problem) Introduction

VRP

- Depot (May have 1 or more), in this paper it simply make every graph has 1 depot.
- Multiple customers, and every customer has their needs.
- Multiple vehicles , in this paper every vehicle has identical capacity.
- **Goals**: To find the paths (may have 1 or more) that maximize / minimize your requests
- Request example: Minimize the distance, minimize the car used, etc.



Source: https://www.mdpi.com/2079-9292/10/24/3147

Definition

- $P0$: Depot
- $P1, P2 \ldots Pn$: Customers / delivery locations
- D: Distance matrix storing distance between each customers,

  $D = [dij \mid i, j = 1, 2, 3 \ldots n]$
- $N1, N2 \ldots Nn$: Needs of each customer
- X: Adjacency matrix, $X = [xij \mid i, j = 0, 1 \ldots n]$
  - $xi0 = 1, i = 1, 2, 3 \ldots n$ (Each customer is connected with $P0$ or at most 1 point)
  - $xij = 0, i = 0, 1, 2 \ldots n$

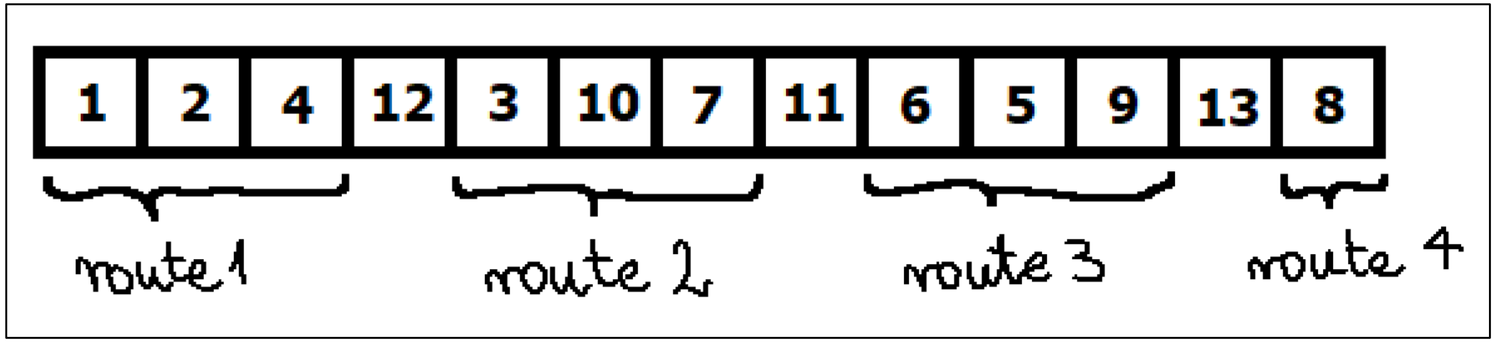
Characteristics

- VRP is **NP-hard** (TSP can be derived into VRP in polynomial time)
- Needs some heuristic algorithm to get near-optimal solution
- Can be **represented using permutation**.

GA and its modifications for the VRP

Representation

- 0: Represent the starting point(depot) in a graph (Can be skipped to increase readablility)
- 1~n: Represent n customers.
- n+1~m-1: Represent m vehicles, where n+1 represent vehhicle 1…, n+k represent vehicle k
- But There are m-n vehicles => 1 vehicle is always skipped (default to use)



Source: https://www.mdpi.com/2079-9292/10/24/3147

Selection

- Tournament selection
- Rank selection
- Roulette wheel selection
- Elitism selection

Crossover

- Partially Mapped Crossover
- Order Crossover (OX)
- Cycle Crossover (CX)
- Edge Crossover

Mutation

- Swap mutation - for simplicity
- It is safe to swap vehicle number and customer number

Conclusion

All combinations, moderate settings, fast experiment

- Vertex-based crossovers provide higher entropy, therefore perform better in short runs.
- Edge-based crossovers perform better, when given enough time. relative to instance size.
- Selection method does not seem to impact final results very much.
- If any selection method were to be chosen, tournament would be the one, as it provides slightly better results than rest.
- The average solution is not optimal but not completely random.
- Better results were obtained in case of smaller examples.
- One thousand iterations is definitely not enough for the GA to even start producing non-random solutions for larger examples.
- Despite harsh conditions for the algorithm, it was still able to find a very optimal solution for one of the examples.

Crossover domination

- Making crossover a dominating genetic operator provides reasonable results.
- Complete lack of mutation results in very bad individuals.
- In general, when the algorithm is run with enough computational resources, more dominant crossover provides better individuals, than more dominant best individuals' copying.
- With tight constraints, crossover domination behaves as badly as more stable approaches.
- When crossover is dominant, at some point the algorithm starts to rapidly descend and find good representatives, but after that rapid inflation process, further improvement is slow.

Mutation domination

- Making mutation a dominating genetic operator provides reasonable results.
- Mutation alone cannot provide results as well as a properly adjusted more stable run with edge crossover.
- Mutation behaves well, when the example has a uniform distribution of destinations.
- On a smaller scale, assigning more computational resources results in a linear decrease in the average results.
- It is recommended to use slightly higher amounts of mutated offspring in the first iteration of the algorithm, as mutation is better than any type of crossover in spreading solutions across the search area.

Best combinations, long, large examples experiment

- Reasonably chosen genetic algorithm parameters can provide enough stability to start reaching satisfying results even for large examples.
- Real life instances pose greater difficulties compared to randomly distributed ones.
- Given enough time and computational power, genetic algorithm implementation can produce competitive results even in the largest known test instances.
- Optimizations of paths in areas with higher density is easier and appears earlier in the whole optimization process than optimization of routes leading to further points of interest.
- Alternating edges crossover combined with rank selection is the most stable choice for genetic operators, capable of finding results comparable with the worldwide best-known ones.