Comparison of Heuristic algorithms for Vehicle Routing Problem

Vojtěch Fiala, Peter Močáry

December 14, 2023

Vehicle Routing Problem

- Finding the best paths to visit all customers and deliver them the requested goods
- NP-hard problem
- It has many variants – Capacitated VRP
- Complete graph with n nodes and $\frac{n(n-1)}{2}$ edges
- Exact and heuristic approaches

Improved Clarke-Wright Savings Algorithm

- Heuristic approach to CVRP
- Savings principle
- Modification of the original Clarke-Wright savings algorithm
- Reasonable results for small to medium size problems
- Time complexity: $O(n^5)$
- Space complexity: $O(n^2)$

Improved Clarke-Wright Savings Algorithm

- Step one: Calculate distances between customers
- Step two: Calculate savings between customers
- Step three: Initialize a new route
- Step four: Chose a candidate customer for the route
  - Accept the customer only if the savings of the connection is the largest possible for the candidate customer.
- Step five: Repeat the Step four until no customer can be added to the route
- Step six: Repeat the Step three, Step four and Step five until all customers are in a route.

Genetic Algorithm

- For small & medium problems
- Uses a population of chromosomes to produce increasingly better solutions
- Fitness method to evaluate solution quality (Euclidean distance)
- Initial population generation (Randomly generated)
- Repeat until specified iteration count is reached:
  - Select two parents (Binary tournament)
  - Produce offspring using crossover (Ordered crossover)
  - Mutate the offspring (Randomly switch two customers on different routes)
  - Replace worst member of the population
- Split the best solution into subroutes

- Time complexity: $O(n^2)$, but constants are crucial here $O(i \cdot (n \cdot (10 + p) + 4n^2))$ – i is the number of iterations and p is the population size
- Space complexity: $O(n)$

Experiments

- 100+ problems, datasets with < 200 nodes
- Overall distance travelled
- Number of vehicles needed
- Number of customers per route
- Number of routes linking only 2 customers
- Total unused vehicle capacity
- Run time

Results for small to medium problems (10-100) and medium problems (100-200) are as follows:

- Savings finds on average **1.18**× and **1.14**× longer total distance
- Savings algorithm needs on average **1.6**× and **3.4**× more vehicles
- Savings serves on average **0.64**× and **0.33**× less customers on each route
- GA had a total of **0** and **79** routes with only 2 customers, Savings had **75** and **455** of them
- Savings has on average **1.08**× and **1.54**× more unused space in vehicles than GA
- GA is on average **2695**× and **8.3**× slower than Savings – in practical terms, this is negligible

Conclusion

- Both algorithms perform best at small and medium problems
- Savings produces 1.16× longer paths.
- GA produces better results overall but at a high price of much worse run-time
- With increasing number of nodes, run-time difference lessens (Savings has much worse time scaling).

Discussion