Overview

- ➢ Objective and Scope
- ➢ Literature Survey
- ➢ Problem Formulation
- ➢ Experimental results
- ➢ Conclusion
- ➢ References

Objective

Our objective in this project is to stimulate and experiment the methods proposed by previous works in VRPTW that is to minimize the number of vehicles and total distance traveled to service the customers without violating the capacity and time window constraints and to explore various areas to get better solutions of The Vehicle Routing Problem with Time Windows (VRPTW) using multi-objective genetic algorithm to develop more refined approach by experimenting with different type of crossover and mutation methodology and rate. A set of well-known benchmark data are used to compare the effectiveness of the proposed method for solving the VRPTW. The scope of project include getting optimal solution for upto hundred customers using Multi-Objective Genetic Algorithm(MOGA) with different number of generations.

Literature Survey

In 1959, VRP first appeared in a paper[3] by George Dantzig and John Ramser formulated as a truck dispatching problem.

Different variations of VRP has been developed over the years.The problem we are working on is a hybrid of Capacitated Vehicle Routing Problem (CVRP) and Vehicle Routing Problem with Time Windows (VRPTW).

Researchers have studied VRPTW using exact and approximation techniques.

Kohl's work [4] is his one of the most efficient and accurate methods of VRPTW. He was able to solve 100 different customer sized instances. However, no algorithm has been developed that can optimally solve all his VRPTWs with more than 100 customers.

Literature Survey(Contd.)

Gehring and Homberger [5] presented a two-stage hybrid search that first uses an evolutionary strategy to minimize the number of vehicles and then uses a tabu search algorithm to minimize the total distance.

Luca Maria Gambardella et al.[6] studied one type of multi-objective implementation of VRPTW by minimizing a hierarchical objective function.Here the first objective was to minimize the number of vehicles and the second is to minimize the total travel time. This was done by adapting the Ant Colony system "ACS". [7] Define two ant colonies, each dedicated to the optimization of each objective function.

Ombuki et al. (2006) presented a multi-objective genetic algorithm for the time window problem of vehicle routes, aiming at minimizing the total number of vehicles and the total cost (distance). When resolving, they used Best Cost Route Crossover (BCRC) as the crossover operator and Constrained Route Reversal Mutation as the mutation operator. They concluded that their approach was effective because the solutions obtained competed with the best-known solutions in the literature.the approach was in two stages: GA was used first to establish the number of vehicles, and in a local tabu search was conducted to minimize the total cost of traveled distance.Essentially, the multi-target VRPTW problem turned into a single-target optimization.

Problem Formulation

What is VRPTW(Vehicle Routing Problem with Time Windows)?

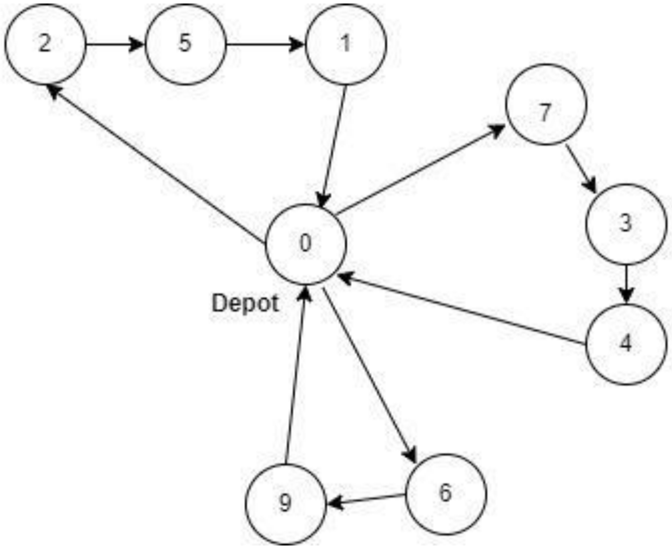
Delivering goods to customers at different locations with each customer having a hard time window and multiple vehicles is used with homogenous capacity.

Scenarios to be tackled

- When capacity of the vehicle exceeds the maximum limit then number of vehicles should be increased.
- When a vehicle arrive before ready time it needs to wait till ready time i.e. hard time window is followed
- When vehicle arrive after due time then a new vehicle is needed and hence distance travelled increases.

Problem Formulation(Contd.)

This problem is a real time adaptation of Travelling Salesman Problem(TSP).Hence It is a NP Complete problem and thus a heuristic and metaheuristic approach would be suitable to get optimal solution.



Problem Formulation(Contd.)

We are working on Multi-Objective Genetic Algorithm approach for this problem.

Genetic Algorithm (GA) is a widely used search method for complex problems such as VRP. Selection, Crossover, and Mutation are the main operators used in GA.

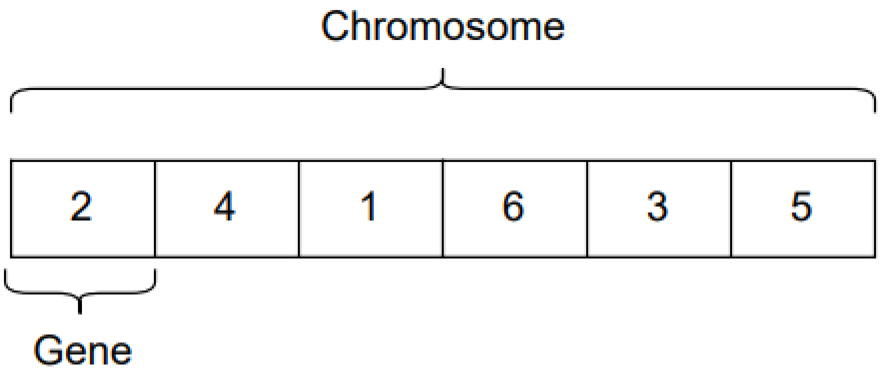
- GENETIC_ALGORITHM(….)
  - INITIAL_POPULATION(….)
  - NEW_GENERATION(….)
    - TOURNAMENT_SELECTION(…)
    - CROSS_PARENTS(..)
    - MUTATE(….)
  - WHILE GENERATIONS :
    - NEW_POPULATION = NEW_GENERATION(….)
  - RETURN GENOTYPE(BEST_CRHOMOSOME(…))

Problem Formulation(Contd.)

Data Structure Used:

Each customer is represented as a gene and an array of customers that will show the routes of the vehicles is chromosome. A chromosome will represent a solution for the problem. A group of chromosomes represented as population.

This example has 6customers



Problem Formulation(Contd.)

Population Generation

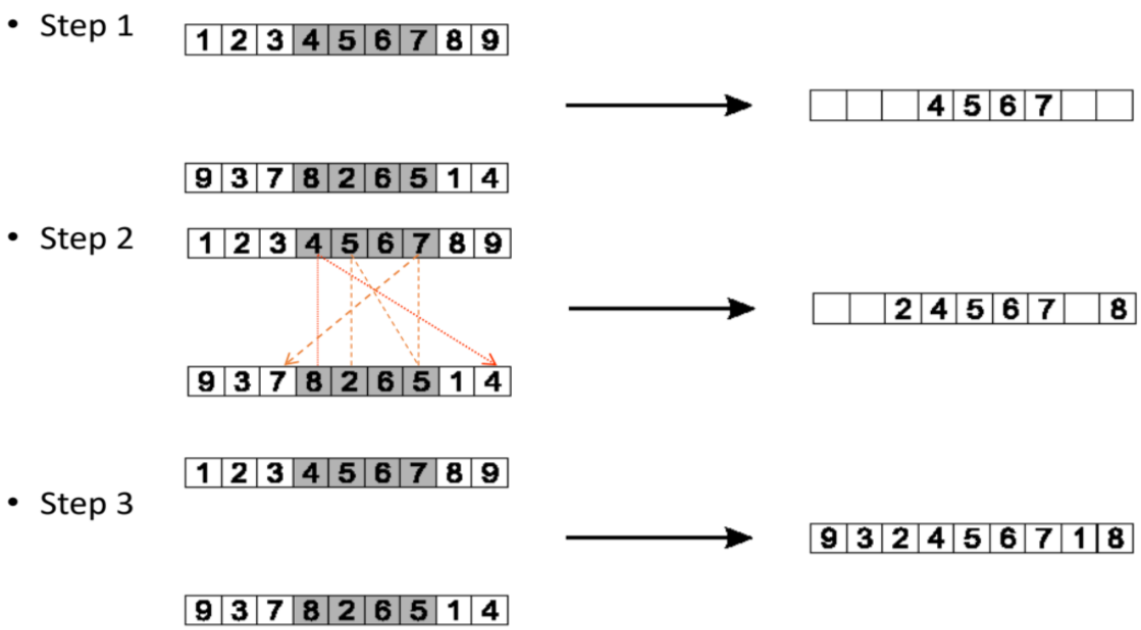
Initial population is generated with population size 100 by selecting random permutation of customers and then GA operators are applied to the population to get further optimized solution with each iteration as shown in diagram of typical Genetic algorithm above.

Crossover

Following crossover has been experimented:

1.PMX (Partially-Mapped Crossover)- Subscripts P1 and P2 (with P1 and P2) are uniformly chosen here at random. Through position-by-position exchange procedures, the two points (P1 and P2) define a matching section that is used to make a cross.

The two sub-routes are typically switched between the crossing points via PMX. There are multiple batch numbers. The two new solutions then undergo a different procedure. If a duplicate batch number in solution 1 is discovered, it is exchanged with the batch number of solution 2 in front of the first duplicate batch location of solution 1 (for example, solution 1's solution 5 is replaced with solution 2's solution 7). For duplicate solution 2 batch numbers, the same process is used. The procedure is repeated until all duplicates are gone.



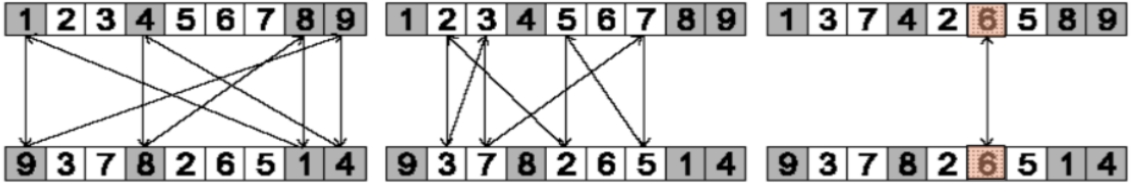
PMX Crossover

Problem Formulation(Contd.)

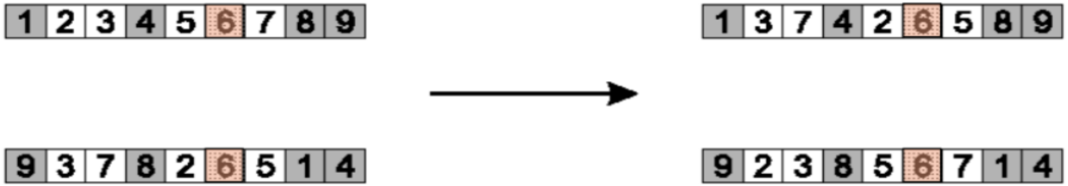
2.Cyclic Crossover: In this crossover, the positions of the genes in both parents are matched. Then a cycle is created by traversing through the genes in one of the parents. The genes in the other parent that are in the same cycle are then copied to the offspring in the same positions.

Cyclic crossover can be useful in maintaining the order of genes that are important for the solution quality and can be applied to various optimization problems such as scheduling, routing, and packing.

- Step 1: identify cycles



- Step 2: copy alternate cycles into offspring



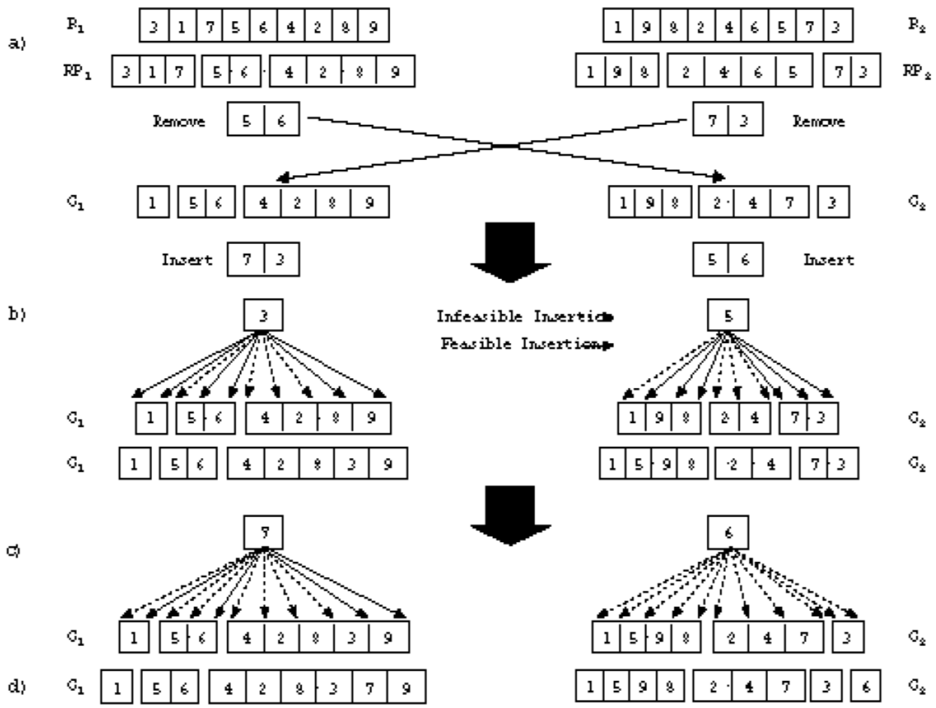
Cyclic Crossover

Problem Formulation(Contd.)

3.Best Cost Route Crossover, BCRC: It aims at minimizing the number of vehicles and cost at the same time while considering feasibility constraints.

In BCRC, two parents are selected randomly lets say P1 and P2, from these two parents some genes are selected from a route and in opposite parent these genes are deleted. Now one by one genes deleted from parents are inserted at best possible locations to create two children C1 and C2 from P1 and P2.

Problem Formulation(Contd.)



BCRC Example Explanation

Problem Formulation(Contd.)

Mutation

Following mutation has been experimented

1.Swap Mutation: It select two positions in the chromosome at random and interchange the values.

1	2	3	4	5	6	7	8	9	0

=>

1	6	3	4	5	2	7	8	9	0

2.Cyclic Mutation: It selects a subset of consecutive genes and cyclically shifts their values.

3.Inversion Mutation: It selects a subset of genes and inverts the entire string in the subset.

0	1	2	3	4	5	6	7	8	9

=>

0	1	6	5	4	3	2	7	8	9

Problem Formulation(Contd.)

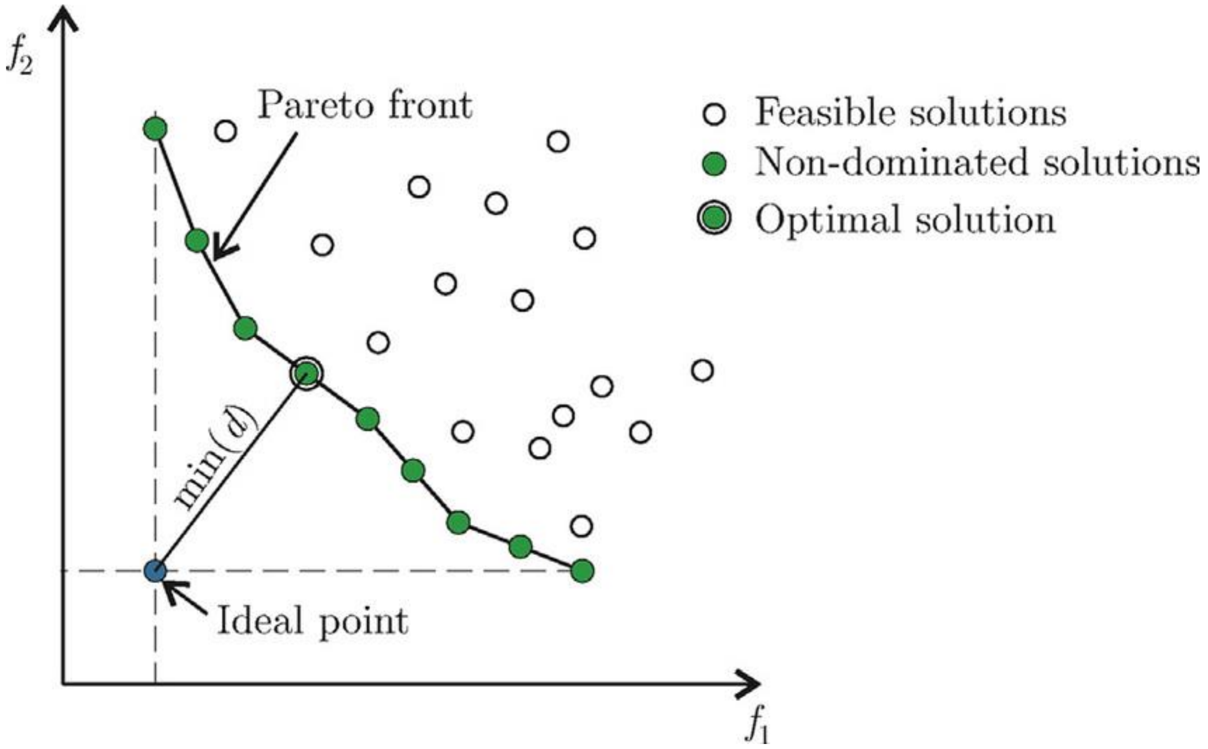
Selection

After crossover and mutation new children are generated and are added to the population.

An elite model is incorporated to ensure that the best talent is passed on to the next generation. The advantage of the elitist method over traditional stochastic reproduction is that it ensures that the current best solution from the previous generation is copied unchanged to the next generation. This means that the best solution produced by the best overall chromosome never degrades from one generation to the next.

We sort the population on basis of pareto ranking and size of population is restored to initial population size. The chromosomes with better pareto ranking are retained and chromosomes with not so good ranking are eliminated.

Problem Formulation(Contd.)



Problem Formulation(Contd.)

Termination

In GA we run our algorithm to a number of generations till a termination condition is met. We are taking number of generations as a terminating condition in our case. For now we take number of generations as 450 for termination condition.

Experimental Results

Execution Environment

The code was written in C++ and executed on a computer with 8GB of RAM and an Intel(R) Core(TM) i5-8265U, 1.60GHz processor.

Experimental Results(Contd.)

Analysis of Dataset

The typical Solomon's VRPTW benchmark [9] problem examples are used in our experimental findings.

Solomon's data is grouped in six classification: **R1,R2,C1,C2,RC1,RC2.**

R ➔ Uniformly distributed customer locations.

C ➔ Clients are grouped either geographically or according to time intervals.

RC➔ Hybrid problem with mixed features from both R and C.

Experimental Results(Contd.)

Experiments and Obtained Results

We have experimented with different combinations of crossover and mutation that have the potential to be used for this scenario i.e. VRPTW problem. The crossover and mutation explained above is experimented and the results obtained are recorded and shown in next slide for a different number of generations.

Experimental Results(Contd.)

Results of experimentation with different Crossover and Mutation:

Number of Generations: 300

	Partially-Mapped Crossover	Cyclic Crossover
Swap Mutation	(52,3433.64)	(54,3426.28)
Cyclic Mutation	(59,3635.18)	(60,3681.06)
Inversion Mutation(3 Elements)	(51,3380.07)	(52,3459.91)

Table: 1

Number of Generations: 1000

	Partially-Mapped Crossover	Cyclic Crossover
Swap Mutation	(44,3216.53)	(48,3242.09)
Cyclic Mutation	(57,3610.5)	(58,3624.11)
Inversion Mutation	(51,3361.34)	(51,3361.4)

Table: 2

Number of Generations: 3000

	Partially-Mapped Crossover	Cyclic Crossover
Swap Mutation	(41,2944.87)	(44,3035.67)
Cyclic Mutation	(57,3594.07)	(54,3674.88)
Inversion Mutation	(50,3333.8)	(51,3333.52)

Table: 3

Experimental Results(Contd.)

Thus we can conclude that swap mutation and inversion mutation are performing better than cyclic mutation and giving some promising results. PMX is performing better than the Cyclic crossover operator, as the Cyclic crossover is distorting the routes to more extent.

As BCRC [11] has been introduced specifically for this problem, it is outperforming PMX and Cyclic crossover. When a crossover is performed on two parents using BCRC the children produced are either better or equal to parents in terms of cost, since we are comparing them with their parents for each iteration.

Experimental Results(Contd.)

The experimentation on BCRC and Fitness Function are based on the set of GA settings listed below:

- population size = 100
- generation span = 450
- crossover rate = 0.80
- mutation rate = 0.10

Experimental Results(Contd.)

We are using the fitness function as a parameter to compare the cost of children to their parents in BCRC crossover and we are using the Pareto ranking method to select the elitist population for the next iteration.

The fitness function is formulated as

Fitness = α*Number_of_Vehicles + β*Distance.

Increasing the value of α prioritizes the number of vehicles whereas increasing the value of β prioritizes the total distance traveled.

Different combinations of α and β have been used to obtain the results in Table below.

In every iteration of GA, we are performing crossover and mutation and then selecting the elitist population for the next generation, but we deliberately add some worse population to the next generation which provides randomness and prevents GA to stuck at local minima. We are adding a worse population in a small proportion (2% to 10%).

Experimental Results(Contd.)

INSTANCE DATA	BEST KNOWN RESULT	EXPERIMENT RESULT
R101	19 1650.8	19 1664.82
R102	17 1486.12	18 1496.35
R103	13 1292.68	14 **1251.21** ↑
R104	9 1007.31	11 1023.66
R105	14 1377.11	16 1396.55
R106	12 1252.03	14 1294.64
R107	10 1104.66	11 1125.44
R108	9 963.99	11 1009.94
C101	10 828.94	**10 828.937** ≈
C102	10 828.94	10 832.342
C103	10 828.06	10 837.746
C104	10 824.78	10 848.37
C105	10 828.94	10 **828.937** ≈
RC101	14 1696.94	16 1703.77
RC102	12 1554.75	14 **1519.19** ↑
RC103	11 1261.67	12 1332.02

Comparison with best published result using Solomon's benchmark problems

Experimental Results(Contd.)

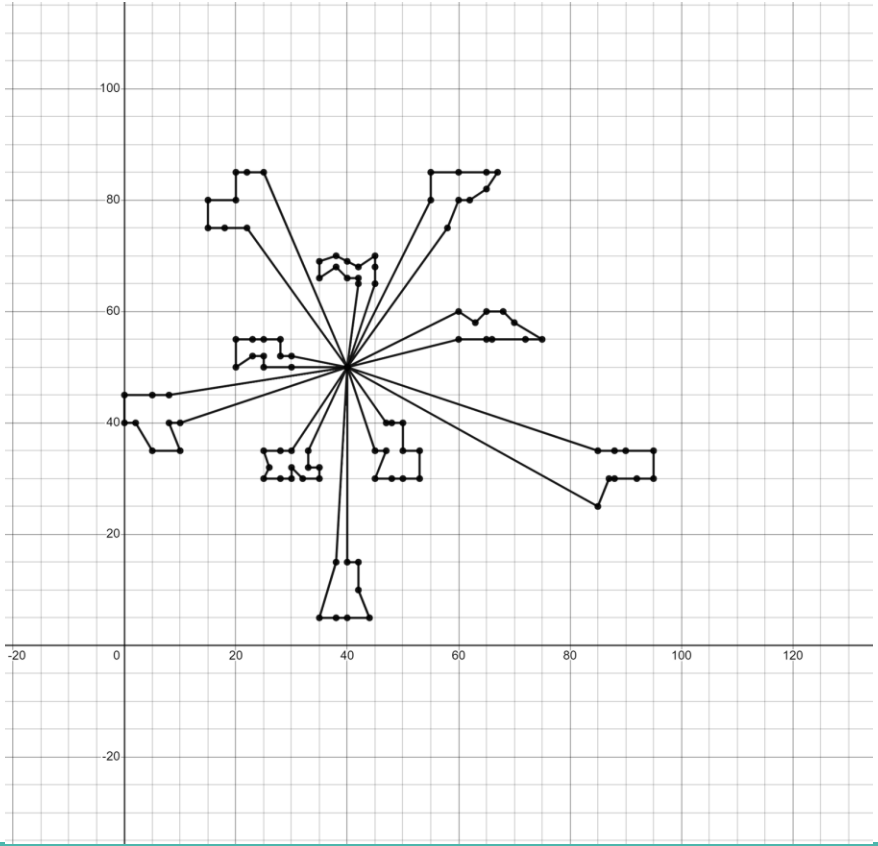
We have compared the results of different datasets using BCRC and inversion mutation (with 3 elements to be inverted). ↑ indicates our results are better (either in terms of distance or in terms of the number of vehicles) than the best-known while ≈ indicating that our results are the same as the best-known results.

Experimental Results(Contd.)

```
Num of vehicle -> 10 Distance -> 828.937
Route 1 ->  4 2 6 7 9 10 8 5 3 1 0 74
Route 2 ->  19 23 24 26 28 29 27 25 22 21 20
Route 3 ->  31 32 30 34 36 37 38 35 33
Route 4 ->  42 41 40 39 43 45 44 47 50 49 51
48 46
Route 5 ->  56 54 53 52 55 57 59 58
Route 6 ->  66 64 62 61 73 71 60 63 67 65 68
Route 7 ->  80 77 75 70 69 72 76 78 79
Route 8 ->  12 16 17 18 14 15 13 11
Route 9 ->  89 86 85 82 81 83 84 87 88 90
Route 10 ->  97 95 94 93 91 92 96 99 98
```

Route Configuration of C101.

Experimental Results(Contd.)



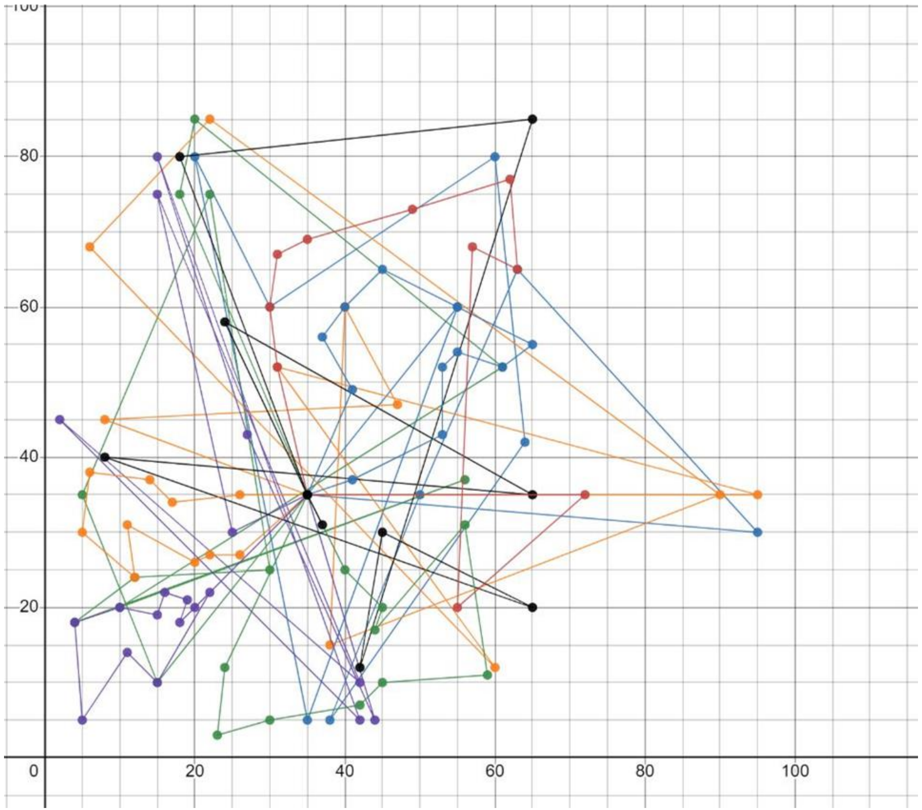
Network configuration for 100 spatially grouped customers across a large time range. Test out issue C101.

Experimental Results(Contd.)

```
Num of vehicle -> 11 Distance -> 1023.66
Route 1 ->  5 4 92 58 94 96 86 56 1 12 57
Route 2 ->  25 71 55 22 66 38 54 3 24 53
Route 3 ->  26 68 75 78 2 28 23 67 79 11
Route 4 ->  17 81 7 47 46 18 6 51
Route 5 ->  87 61 10 62 63 48 35 45
Route 6 ->  49 50 70 34 64 65 31 89 9 30
Route 7 ->  0 69 29 19 8 33 77 80 32 76 27
Route 8 ->  41 42 14 40 21 74 73 72 20 39
Route 9 ->  93 95 98 83 60 16 44 82 59 88
Route 10 ->  91 13 43 37 85 15 90 84 97 99 36
Route 11 ->  52
```

Route Configuration of R104.

Experimental Results(Contd.)



network configuration for 100 evenly-spaced customers with a limited time window

Experimental Results(Contd.)

Depending on whether the customer wants the greatest solutions for trip costs or the best number of vehicles, there are two (or more) solutions when utilizing the Pareto ranking. There may be a single Pareto solution in some experiments that is best known in both the vehicle and distance dimensions, such as experiment c101. Other approaches, like r103, shorten the distance but at the expense of introducing more vehicles.

Conclusion

We have experimented with the different parameters used in the genetic algorithm which focuses on solving the VRPTW with MOGA. This paper constitutes the experimentation done with different crossover and mutation operators and the difference between their results. Our results are comparable to other vehicle-biased results in the literature. We have used a variation of BCRC that is suggested in [11]. The result from data instance C101 is the same as the best-known result. And some of the data instances like R103 and RC102 have shown optimized results in terms of one of the variables i.e. either number of vehicles or distance.

References

1. M. R. Garey, and D. S. Johnson, *Computers and Intractability, A Guide to The Theory of NP-Completeness*, W. H. Freeman and Company, 1979.
2. J. K. Lenstra and A. H. G. Rinnooy Kan, “Complexity of Vehicle Routing Problem with Time Windows,” *Networks*, 11:221-227, 1981.
3. http://andresjaquep.files.wordpress.com/2008/10/2627477-clasico-dantzig.pdf
4. Niklas Kohl. “Exact Methods for Time Constrained Routing and Related Scheduling Problems,” *PhD Thesis*, Department of Mathematical Modeling,Technical University of Denmark, 1995.
5. H. Gehring and J. Homberger, “Parallelization of a Two-phased Metaheuristic for Routing Problems with Time Windows, ” Asia- Pacific Journal of Operational Research 18, 35-47, 2001.
6. L. M, Gambardella, E. Taillard and G. Agazzi, “MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows,” In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, pp.63-76, McGraw-Hill, London, 1999.
7. M. Dorigo, L. M. Gambardella, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,” *IEEE Transactions on Evolutionary Computation* 1:53-66, 1997.
8. https://github.com/raunaktrip/Btech_project_Moga_vrptw
9. http://w.cba.neu.edu/~msolomon/problems.htm
10. http://w.cba.neu.edu/~msolomon/r1r2solu.htm
11. https://www.researchgate.net/publication/227309327_Multi-Objective_Genetic_Algorithms_for_Vehicle_Routing_Problem_with_Time_Windows

Thank You!