AN ENHANCED GENETIC ALGORITHM BASED APPROACH TO VEHICLE ROUTING PROBLEM

CS 255 Project Report

Presented to

Prof. Katerina Potika

Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the requirements for the class

CS 255

By

Akash Janardhan Srinivas

Sumukh Naveen Aradhya

Nov 2023

TABLE OF CONTENTS

I. Abstract.. 3

II. Introduction.. 4

III. Methodology and Implementation..5

IV. Technical Flow.. 7

V. Results... 9

VI. Dataset Used.. 9

VI. Conclusion... 10

VI. Possible Direction For Future Research.. 10

VI. Instructions On How To Compile And Execute The Code.. 10

References.. 11

ABSTRACT

In the current technological realm of computational problem-solving, the design and analysis of algorithms have proven to be the foundation for innovation and efficiency. Our project targets an evident real-world problem in the domain of delivery, logistics, and transportation - Vehicle Routing Problem (VRP), and tries to find better ways to solve it. VRP is about figuring out the best routes for vehicles to save time and total cost by making sure the distance traveled is minimal. We have provided a comprehensive comparison of three different approaches: the Standard Genetic Algorithm, a hybrid enhanced Genetic algorithm with updates in the population creation and convergence functions, and Google OR-Tools. In the world of algorithms, the defining factor is the efficiency and optimality of an approach. Hence, a report like ours presents in-depth insights into various approaches that can be taken to solve a common real-world problem i.e., the vehicle routing problem.

Python along with the DEAP (Distributed Evolutionary Algorithms in Python) library was used as a framework on top of which our version of the Genetic Algorithm was implemented. These algorithms are like nature's way of selecting the best traits for survival, but here they select the best routes. Two techniques namely the sweep approach [2] in the initial population creation and neighborhood search optimization during the convergence of GA were used to make impactful changes to the final output routes. Lastly, it is important to note that we have used the Google OR-Tools as a baseline for comparison assuming it to be the optimal result for a given distance matrix.

Our findings showed that each method has its strengths and weaknesses. The Pure Genetic Algorithm was good at searching for solutions but needed several enhancements to reach near-optimal results. The enhanced Genetic Algorithm was a simple and effective approach that led to significant improvements in these solutions.

Our project helps readers understand how various methods can be utilized to solve the Vehicle Routing Problem. Delivery companies and businesses could heavily benefit from these insights and solutions by utilizing them to decide the best route for their transportation and delivery needs.

INTRODUCTION

The Vehicle Routing Problem (VRP) is vital in the world of delivery and transportation. Consider this scenario: a delivery company needs to send a bunch of trucks to deliver packages to different places. VRP is used to figure out the best way for these trucks to be scheduled so that they can deliver using the least amount of time and the least amount of fuel. This looks simple, but it gets tricky when we have a large number of trucks, several delivery locations, and various constraints such as how much load a truck can carry and what is the time window within which a delivery has to be made. We aim to understand and implement different approaches available where Genetic Algorithms can be employed to solve VRP contrast three different ways this can be achieved and also evaluate which approach works best. These approaches are:

Randomized(Pure) Genetic Algorithm: This is an interesting way of solving problems that is similar to how nature works. It comes under the class of evolutionary algorithms and it parses several solutions and keeps making them better over time, similar to how humans and animals evolve

Hybrid Enhanced Genetic Algorithm: This method builds on top of the basic GA, removing some randomness and replacing it with semi-structured population creation, and utilizes a neighborhood search algorithm, specifically a two optimal search algorithm during the convergence phase, to get more accurate and near-optimal results. It takes a route and keeps making several changes to see if the result can be improved

Google's OR Tools: This is a set of tools made by Google that are good at solving VRP problems. This has been considered optimal and is being used as a baseline to compare the above-mentioned algorithms

By comparing these methods, we want to find out not just which approach is the best at solving VRP but also understand a bit about why they are good for different kinds of delivery and route problems. This way, companies or anyone who needs to plan routes can pick the best method for their needs.

METHODOLOGY AND IMPLEMENTATION

Genetic Algorithms are a part of evolutionary algorithms. They mimic the process of natural selection. It's like survival of the fittest but for computer algorithms. In these algorithms, a 'population' is created of possible solutions to a particular problem, in our case the Vehicle Routing Problem. These solutions evolve over generations and become more efficient after each iteration.

How we have utilized GAs to solve VRP:

- **Chromosome and Gene**: The simplest unit in genetic algorithms is a gene, here in VRP a gene is an individual city that a vehicle is associated with. A chromosome is a possible solution to the VRP problem, which has the order in which the cities have to be visited from the depot along with the vehicle allocation
  Example chromosome : [[10, 1, 4, 6, 7, 3], [2, 8, 9, 5, 11]]

- **Population:** We started by generating a random population of routes and our initial population had a thousand chromosomes

- **Fitness Function:** This function evaluates how good a route is. It is based on the total distance and feasibility of the solution

- **Selection:** In each generation, the fittest routes are selected, meaning those routes that had the shortest total distance. This is done using a tournament approach. If a tournament size is three then it means that three chromosomes among the initially chosen thousand are picked and the best among the three are propagated to the next generation. This process is repeated a thousand times where during each iteration the algorithm picks the thousand best-evolved chromosomes for the next generation

- **Crossover**: We combine parts of two good routes to create a new route hoping that it inherits the best qualities of both. Here the two initial routes are called the parents and the combined routes are called offsprings. This crossover uses a random function that decides the indices at which the chromosome will be cut. The crossover is done between the cuts that give rise to a new chromosome

- **Mutation:** This step is important for introducing variety and avoiding similarity. We have slightly altered some of the routes randomly. Mutation is being done in two different ways here:
  Swapping: Two random genes are selected and swapped, to give rise to a new chromosome
  Shuffling: Two indices are selected and the genes between these indices are randomly shuffled to form a new chromosome

All the techniques and processes mentioned above are part of a Randomized Genetic Algorithm. We have incorporated three different approaches to efficiently showcase the comparison of how the routes for a VRP vary based on different kinds of approaches utilized. We begin by running an open-source algorithm by Google called ORTools that solves the vehicle routing problem. We have considered this to be an optimal solution as it yields the best result (least distance) among the algorithms we have executed. According to [1], the authors have elaborated on the usage of techniques like Tabu search or simulated annealing to solve VRP to obtain optimal results. Considering these optimization techniques are research topics of their own, we have considered a simpler approach using Google ORTools to be an optimal solution for our model evaluation. OR-Tools is a software suite developed by Google that's great for problems like VRP.

How We Used OR-Tools in VRP:

- **Model Setup:** We defined our VRP in OR-Tools, including the number of vehicles, their capacities, and the distances between different locations
- **Solver Configuration:** We configured the solver parameters, setting it to minimize the total distance
- **Solution Extraction:** After running the solver, we extracted the routes it suggested for our VRP and compared it with the results GA and hybrid GA models are producing

We have then implemented a Pure Genetic Algorithm Vehicle Routing Solution that uses randomization in every genetic algorithm step (elaborated in the previous paragraphs). This algorithm yielded results that were quite far from the optimal solution (greater distance traveled).

Finally, we have considered enhancement techniques as suggested in [1], wherein we have updated the initial population creation function to consider cities in the increasing order of their polar angle concerning the depot (as elaborated by authors of [2] known as sweep approach), instead of randomly choosing a population. Next, we have updated the way the GA is converging by aiding it to accelerate faster than what is expected. This is being achieved by utilizing a neighborhood search algorithm, specifically a two-optimal search algorithm where we take a route, consider a path from depot to two cities in the given route, reverse its path, and connect the edges to check if we can get a shorter path. If we find a shorter path then we update the route and move forward with further execution.

Implementation of Genetic Algorithms (GAs)

- **Setting up the Environment:**
  We began by setting up Python and importing necessary libraries such as DEAP for genetic algorithms, NumPy for numerical operations, and Matplotlib for visualization.
- **Evolutionary Process:**
  The GA's evolutionary process had to select the best routes from the population (selection), combine them to create new routes (crossover), and introduce random changes (mutation) to maintain diversity.
  This process was iterated over 1000 generations, with each generation producing fitter routes than the previous one and finally converging on an optimal route

TECHNICAL FLOW AND ALGORITHM

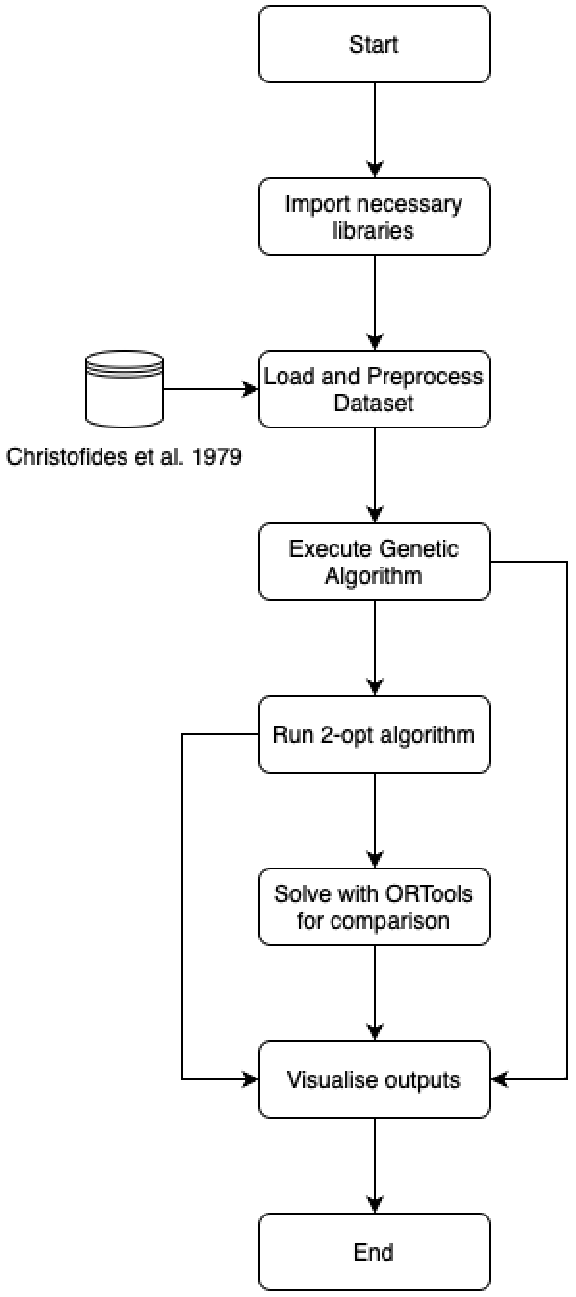
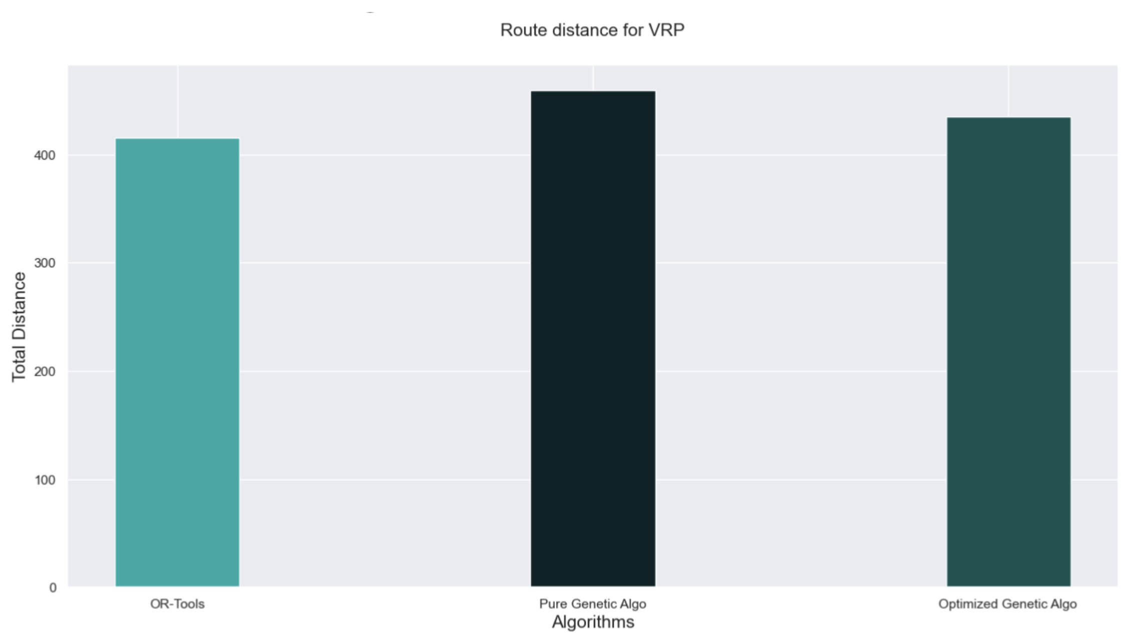

Start
Import necessary libraries
Load and Preprocess Dataset
Christofides et al. 1979
Execute Genetic Algorithm
Run 2-opt algorithm
Solve with ORTools for comparison
Visualise outputs
End

Fig 1. Technical Flow Diagram

Algorithm:

1. Import necessary libraries: Streamlit, NumPy, Pandas, Matplotlib, Seaborn, SciPy, DEAP, OR-Tools
2. Load and preprocess dataset:
    a. Parse data into a python dataframe
    b. Remove columns that are not necessary for our case (demand)
    c. Remove redundant rows if any
3. Define plot and problem parameters, necessary data structures and the DEAP toolbox, creator and fitness functions
4. Run randomized genetic algorithm (randomization involved in each step):
    a. Generate an initial population (randomization technique and sweep approach [2] - increasing order of polar angle)
    b. Evaluate fitness of the solution
    c. Extract routes from the solution
    d. Calculate the total cost of a route
    e. Perform crossover between two solutions
    f. Get random cut points for crossover or mutation
    g. Swap genes between two solutions
    h. Perform mutation on the solution, either swap or shuffle (with 0.5 probability)
    i. Ensure that vehicle payload constraints are met (unit payload considered in our case) and evaluate population fitness
5. Run the 2-opt algorithm [1]
    a. Select parents for the offspring using tournament selection
    b. Create offspring through crossover and mutation operations
    c. Ensure feasibility of the offspring by adjusting routes if necessary
    d. Evaluate the fitness of the offspring
    e. Update the population with the offspring
    f. Track and update the best solution found so far
6. Solve the problem using ORTools
7. Plot the comparison graph, extract and display the best routes

RESULTS



OR-Tools	415.48 miles
Pure Genetic Algo	459.71 miles
Optimized Genetic Algo:	434.87 miles

DATASET USED

We have used the Christofides et al. 1979 (VRP-REP) dataset. This belongs to a large class of variants of vehicle routing problem datasets. We have considered a simple single depot vehicle routing problem, and have used a dataset with 27 cities for easier representation and analysis. The same has also been run on a dataset with 121 cities and we have obtained similar results. The dataset contains 3 columns, the x-coordinate of the city, the y-coordinate of the city, and the demand value, where we consider only the x and y coordinates of cities and drop the demand column as we are not working with a capacitated vehicle routing problem.

CONCLUSION

We used 3 approaches to solve the Vehicle Routing Problem. ORTools was considered to be optimal and used as the baseline to compare the genetic algorithms. Randomized genetic algo performed quite well giving results within 10% of the optimal solution. The hybrid enhanced genetic algorithm had two major changes, first initial population based on increasing order of polar angle [2] sweep approach, and the second using a neighborhood search algorithm, specifically a two-optimal search algorithm to accelerate GA convergence, this gave within 5% of the optimal results which was much better than the randomized genetic algorithm.

POSSIBLE DIRECTION FOR FUTURE RESEARCH

This research opens up numerous possibilities for future exploration. We have focused on specific aspects of VRP in which the field itself is very vast. It has the potential for integrating other optimization techniques and exploring more complex variants of VRP. Additionally, the incorporation of constraints like time windows, vehicle capacities, and real-time traffic data could enhance the applicability and effectiveness of these algorithms in practical scenarios.

INSTRUCTIONS ON HOW TO COMPILE AND EXECUTE CODE

Step 1: Run the requirements.txt file using the command: ‘pip install -r requirements.txt’. This file installs all necessary packages in order to run the code
Step 2: Update the path for the dataset in vrp.py line number 41, to the location where the dataset is present
Step 3: Run the command, ‘streamlit run vrp.py’. This will open up the UI on your localhost
Step 4: Interact with the UI. By default, the Google ORTools algorithm will run and provide a result on the UI. Next, pressing the ‘Run GA models’ button will sequentially run the pure GA model then the hybrid enhanced GA model, and then provide a graph showcasing the comparison of how pure GA and hybrid GA models have performed against the results obtained by Google ORTools

REFERENCES

[1] Barrie M. Baker, M.A. Ayechew, ‘A genetic algorithm for the vehicle routing problem, Computers & Operations Research’, Volume 30, Issue 5, 2003, Pages 787-800, ISSN 0305-0548,

[2] Billy E. Gillett, Leland R. Miller, (1974) A Heuristic Algorithm for the Vehicle-Dispatch Problem. Operations Research 22(2):340-349.

[3] Braekers, Kris & Ramaekers, Katrien & Nieuwenhuyse, Inneke. (2015). The Vehicle Routing Problem: State of the Art Classification and Review. Computers & Industrial Engineering.

[4] T. Mansur Fantazzini and M. T. Arns Steiner, "Vehicle Routing Problem Solved through Heuristics Procedures: A Case Study," in IEEE Latin America Transactions, vol. 17, no. 05, pp. 858-864, May 2019, doi: 10.1109/TLA.2019.8891955.

[5] **Dataset:** http://www.vrp-rep.org/datasets/item/2017-0027.html