Optimization of the vehicle routing problem using Artificial Bee Colony algorithm

Nikhil Haresh Keswaney
nhk8621@g.rit.edu

***Abstract*—Capacitated Vehicle Routing Problem(CVRP) is a combinatorial problem. CVRP is a NP-Hard problem and the goal of the problem is that for a given set of customers with demands and a fleet of vehicles with capacities, find the most optimal route for all the vehicles such that the demands of all the customers are satisfied. This paper presents multiple approaches for solving the capacitated vehicle routing problem. It first explains an exact approach which combines the subset partition problem and the traveling salesman problem, then an approximate algorithm which is the Artificial Bee colony(ABC) algorithm. In the end a comparative study comparing and contrasting the runtime of all the approaches on datasets of various sizes is presented.**

I. Introduction

The vehicle routing problem (VRP) is a combinatorial problem, which has a wide spread of applications like transportation, e-commerce etc. It appeared first in 1959 in a paper written by George Dantzig and John Ramser [3]. The goal of Vehicle Routing Problem(VRP) is to serve N number of customers using K number of trucks minimizing a cost function. The cost function can be anything like the distance travelled, fuel utilization, number of left turns etc. There are many variants of the vehicle routing problem such as vehicle routing problem with time windows, dynamic vehicle routing problem, capacitated vehicle routing problem etc [4]. The problem that this paper tackles is the capacitated vehicle routing problems(CVRP).
There has been extensive research carried out on VRP due to the importance of the problem and there are multiple ways to solve the VRP which include:

1) Exact algorithms that always return the optimal value
2) Approximate algorithm which doesn't necessarily return the optimal value but returns near optimal results

The problem with using the exact algorithm is the computational complexity. Doing a brute force algorithm on a processor having ~ 1GHz of clock cycle and computing the optimal path for ~ 50 customers takes $\sim 9.6 \times 10^{47}$ years [5], which is not affordable. However, there are certain fast exact algorithms like, branch and bound, cutting plane which reduces the the computation time, but they also have a very high computational complexity especially in large VRP problems. Therefore, approximate algorithms have become popular in recent times [6]. Even though the approximate algorithms don't return the most optimal solution they complete in a reasonable amount of time. Some approximate algorithms that have been used to solve CVRP are tabu Search, ant colony optimization, genetic algorithm, assembly-based memetic algorithm, simulated annealing etc [7].

A. Problem statement

The capacitated vehicle routing problem is a combinatorial problem which states that there are N customers for whom goods have to be delivered using a fleet of vehicles utilizing the most optimal route and each vehicle in the fleet needs to start and end at the same depot. The state space can be represented using a graph.

$$G \equiv \{V, E\}$$

There are N customers that need to be served and can be represented by $S \equiv \{1, 2, 3 \dots N\}$
And since all vehicles start and originate from the same depot assuming it is 0, we get:

$$V \equiv \{0\} \cup S$$

The vehicles are denoted as $K \equiv \{1, 2 \dots k\}$ and the vehicles capacity will be denoted by q_k which will be the same for all the vehicles.
Each customer has a demand and the demand of ith customer is denoted by d_i. Each edge i.e. the path for going from customer i to customer j is represented by c_{ij}. Additionally, we have to satisfy certain other constraints to solve the problem like:

1) Finding the most optimal route minimizing the cost of travel
2) All vehicles will start from the depot and end at the depot.
3) One customer will be served once by only one vehicle.
4) The vehicles the visit the customer will also serve the customer.
5) The sum of the demands for the customer that one vehicle visits shouldn't exceed the capacity of the vehicle.

II. dataset

The dataset used for this paper is the vehicle routing problem (VRP) dataset from BranchAndCut.org [8]. It has ~ 107 files for CVRP problem.

III. Exact algorithm

The exact algorithm for the problem is divided into two parts. The first part will be to find all possible ways to divide the customers into K subsets where K is equal to the number of vehicles in the fleet and the second part would be to find the the best subset using the travelling salesman(TSP) algorithm. The first part returns all the possible ways that the customers can be divided and served by K vehicles and then the best amongst these is chosen by using the traveling salesman algorithm.

A. Subset finder

We have a set of customers or nodes which is denoted by $S \equiv \{1, 2, 3 \dots N\}$ we have to partition S into P in such a way that

$$A \cap B = \emptyset$$

$$A, B \in P$$

The approach used was a dynamic programming approach. The number of partitions the set will be divided will be the number of vehicles that will be delivering the items. Assuming there are K vehicles with capacity q_k that will be delivering the goods. The recurrence relation for the problem will be as follows.

$$s(n, k) = k \times s(n-1, k) + s(n-1, k-1)$$

where, s(n,k) denotes all the partitions that are possible while dividing the set n into k sets [9] The above recurrence relation should follow the following constraint:
Let C be a candidate subsets in all the possible subsets of $s(n, k)$ and J be the path for a vehicle in C and each customer in J will have a demand d_i then $\sum d_i \leq q_k$

B. Traveling salesman problem

The traveling salesman problem states that given a set of vertices, we need to minimize the cost for the route such that all the vertices are visited once. The approach used in this paper for implementing the traveling salesman problem is backtracking. We are assuming that there are N customers and the cost for going from customer i to customer j is c_{ij}. Assuming $C(S, i)$ represents the minimum cost path to visits all the nodes in S starting from the depot and ending at i. So the recurrence relation would be as follows.

$$C(S, i) = \forall i \in S \min\{C(S - \{i\}, j) + c_{ij}\}$$

IV. Artificial Bee Colony Algorithm

A. Description

Artificial bee colony algorithm is based on the foraging behaviour of honey bees. It is a swarm intelligence algorithm which is based on three main components decentralization, self-organization and collective behaviour [**?**]. Since there is no one central entity that controls the whole algorithm and each individual is responsible for its own action, self-organization means that the set of rules for each individual will lead to a neat organization of the entire swarm and collective behaviour is the task of finding the best food source which is done collectively.

The Artificial bee colony algorithm is made of three type of bees which are as follows:

1) Scout bees: The job of the scout bees is to go and find new potential neighbourhood for food sources.
2) Employed bees: around 50 % of the swarm bees are employed bees. The job of the employed bees is to explore the neighbourhoods found by scout bees.
3) Onlooker Bees: The onlooker bees look over the work that is performed by the employed bees and exploit food sources in the neighbourhood of the employed bee that has the richest food source.

B. Working

In the beginning the scout bees go out and find possible food sources i.e. neighbourhoods where food with high nectar can be available, then it comes and informs the employed bees. Then out of these neighbourhoods the employed bees will select a neighbourhood and start exploiting the food sources in the neighbourhood only on employed bee will search in one neighbourhood. While exploiting the food sources the bees will also try and find other food sources in the neighbourhood if they find a food source which is better than the current food source they will start exploiting that food source and abandon the previous food source. Once the employed bees are done exploiting the food sources the bees will move to the dancing area. In the dancing area the onlooker bees will look at the dance of the employed bees and based on the dance they will decide which bee's neighbourhood has better food sources and the onlooker bees will go to that neighbourhood and start exploiting the food sources there if they find a better food source than the employed bee then the onlooker bee will become an employed bee form there on and the employed bee of that neighbourhood will become a scout bee. If a neighbourhood in which the employed bees are searching for food depletes i.e. the employed bees aren't able to find any food source in the neighbourhood then the employed bees abandons the neighbourhood and become scout bee to find a new food source. The whole process is repeated a fixed number of times and the neighbourhood with the best food source is remembered.

C. Fitness function

This is one of the most important part of the artificial bee colony algorithm. The fitness function is the measure of how good the food source is in the neighbourhood it is the nectar quantity of the food source. [1]

$$f(x) = c(x) + \beta * p(x)$$

where,
$p(x) = \sum_{i=1}^{N} d_i y_{ik} - q_k$

D. Implementation

1) *Scout Bee Phase*
In this section of the algorithm the scout bees are utilized to find the initial solutions. For this we randomly shuffle the customers and then divide the customers in trucks according to the truck capacity so there are cases in which sometimes the constraints cannot be satisfied i.e. infeasible solutions which can be used as initial solution but the rule selected in this paper that changes the performance of the algorithm significantly is that we always use feasible solution so we get an infeasible solution we discard it and continue to find a solution until we find a feasible solutions in the search space.
2) *Employed Bee Phase*
In this phase the employed bees go out and explore the food source found by the scout bee. The employed bee then tries to find a better food source in the neighbourhood. So to generate neighbours we use the following operations.
   a) Swap Operation: In this operation we select two random trucks out of those two random trucks we select a random customer in both truck and then we exchange the customer in these trucks. Like in the above example we selected truck 1 and 3 and in truck 1 we selected customer 2 and in truck 3 we selected customer 7 and we swapped it now truck 1 will serve customer 7 and truck 3 will serve customer 2.

	Truck1					Truck2				Truck3			
Before Swap	0	1	2	3	0	4	5	6	0	7	8	9	0
After Swap	0	1	7	3	0	4	5	6	0	2	8	9	0

Figure 1. Vectors

   b) BMX Operation: In this operation we try to maintain the best part of the candidate solution and shuffle the rest and then divide them into trucks. In the below example truck 2 has the lowest cost so it is maintained and the rest elements are shuffled.

	Truck1					Truck2				Truck3			
Solution	0	1	2	3	0	4	5	6	0	7	8	9	0
shuffled	0	5	2	3	~~0~~	1	~~4~~	9	0	6	8	~~7~~	0
BMX	0	2	3	1	0	4	5	6	0	9	8	7	0

Figure 2. Vectors

We find the neighbourhood using these operators and update if the neighbourhood is better if we are not able to find a better neighbourhood after the limit is exceeded. Then the neighbourhood is abandoned by the employed bees. [1]

$$limit = SN \times K$$

where,
SN = Number of bees
K = Number of trucks

3) *Onlooker Bee Phase*
In this phase the employed bees return with the best solution in their neighbourhood and the onlooker bees use the roulette wheel selection method to select an employed bee's neighbourhood and start exploiting that neighbourhood. If they find a better solution the onlooker bee becomes the employed bee and participate in the employed bee phase.

Following is the flow chart of the implementation.

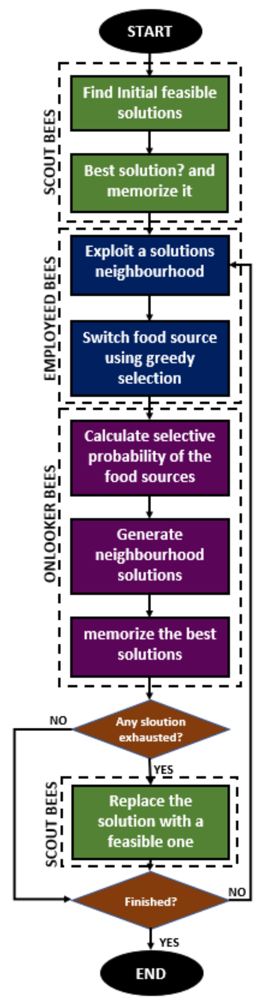

Figure 3. Vectors

References

[1] S.Z. Zhang ; C.K.M. Lee "An Improved Artificial Bee Colony Algorithm for the Capacitated Vehicle Routing Problem" 2015 IEEE International Conference on Systems, Man, and Cybernetics

[2] Harikrishna Narasimhan "Parallel Artificial Bee Colony (PABC) Algorithm" 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)
[3] George Dantzig and John Ramser.
[4] A Survey on the Vehicle Routing Problem and Its Variants
[5] https://medium.com/opex-analytics/opex-101-vehicle-routing-problems-262a173f4214
[6] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet, "A guide to vehicle routing heuristics," Journal of the Operational Research Society, vol. 53, pp. 512-522, 2002.
[7] An Improved Artificial Bee Colony Algorithm for the Capacitated Vehicle Routing Problem
[8] https://www.coin-or.org/SYMPHONY/branchandcut/faq.htm
[9] http://www.sfu.ca/ mdevos/notes/comb_struct/stirling.pdf