Project/Thesis No.: CSER-M-16-02

Solving Capacitated Vehicle Routing Problem through Clustering with Variant Sweep Algorithm and Route Optimization using Swarm Intelligence

By

Zahrul Jannat Peya

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Computer Science and Engineering

Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
May, 2016

Declaration

This is to certify that the Thesis work entitled "Solving Capacitated Vehicle Routing Problem through Clustering with Variant Sweep Algorithm and Route Optimization using Swarm Intelligence" has been carried out by Zahrul Jannat Peya in the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh. The above thesis work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

Signature of Supervisor	Signature of Candidate
Dr. Muhammad Aminul Haque Akhand	**Zahrul Jannat Peya**
Professor	Roll: 1307503
Dept. of Computer Science and Engineering	Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology	Khulna University of Engineering & Technology

Approval

This is to certify that the thesis work submitted by Zahrul Jannat Peya entitled "Solving Capacitated Vehicle Routing Problem through Clustering with Variant Sweep Algorithm and Route Optimization using Swarm Intelligence" has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Department of Computer and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh in May, 2016.

BOARD OF EXAMINERS

1. Dr. Muhammad Aminul Haque Akhand — Chairman (Supervisor)
   Professor
   Dept. of Computer Science and Engineering
   Khulna University of Engineering & Technology
   Khulna, Bangladesh

2. Dr. Muhammad Sheikh Sadi — Member
   Head & Professor
   Dept. of Computer Science and Engineering
   Khulna University of Engineering & Technology
   Khulna, Bangladesh

3. Dr. K. M. Azharul Hasan — Member
   Professor
   Dept. of Computer Science and Engineering
   Khulna University of Engineering & Technology
   Khulna, Bangladesh

4. Dr. Kazi Md. Rokibul Alam — Member
   Professor
   Dept. of Computer Science and Engineering
   Khulna University of Engineering & Technology
   Khulna, Bangladesh

5. Dr. Md. Monirul Islam — Member (External)
   Professor
   Dept. of Computer Science and Engineering
   Bangladesh University of Engineering & Technology
   Dhaka, Bangladesh

Acknowledgements

At first, I would like to thank Almighty Allah for showering all his blessings on me whenever I needed. It is my great pleasure to express my indebtedness and deep sense of gratitude to my Supervisor Dr. Muhammad Aminul Haque Akhand, Professor, Dept. of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET) for his continuous encouragement, constant guidance, great and enthusiastic support, both during the implementation phase and the writing process and keen supervision throughout the course of this study. His prudence and ethics have been a great inspiration for the work. I would like to show my heartiest gratitude to Dr. Muhammad Sheikh Sadi, Professor and Head of the CSE department for his encouragement and support. I also like to remember the inspiration, supports and encouragement of my loving family.

I am extremely grateful to all the faculty members of the Dept. of CSE, KUET to have their privilege of intensive, in-depth interaction and suggestions for the successful completion of my master degree.

May, 2016 Author

Abstract

Capacitated Vehicle Routing Problem (CVRP) is a real life constraint satisfaction problem in which customers are optimally assigned to individual vehicles (considering their capacity) to keep total travel distance of the vehicles as minimum as possible while serving customers. Various methods are used to solve CVRP in last few decades, the most popular way of solving CVRP is splitting the task into two different phases: firstly, assigning customers under different vehicles and secondly, finding optimal route of each vehicle. Sweep clustering algorithm is well studied for clustering nodes. On the other hand, route optimization is simply a traveling salesman problem (TSP) and a number of TSP optimization methods are applied for this purpose. This study investigates a variant of Sweep algorithm for clustering nodes and different Swarm Intelligence (SI) based methods for route generation to get optimal CVRP solution. In conventional Sweep algorithm, cluster formation starts from 0^0 and consequently advance toward 360^0 to consider all the nodes. In this study, a variant Sweep cluster is investigated from different starting angle. A heuristic based adaptive method is developed to select cluster formation starting angle. On the other hand, two well-known optimization methods (i.e., Genetic Algorithm and Ant Colony Optimization) and two recent SI based algorithms (i.e., Producer-Scrounger Method and Velocity Tentative Particle Swarm Optimization) are considered for route optimization. The experimental results on a large number of benchmark CVRPs revealed that different starting angles have positive effect on Sweep clustering and finally, VTPSO is able to produce better solution than other SI methods. Finally, the proposed mythology is found to achieve better CVRP solutions for several problems when compared with several prominent existing methods.

Contents

List of Tables

List of Figures

Nomenclature

SS	Swap Sequence
SOs	Swap Operators
PS	Partial Search
ST	Self-Tentative
GA	Genetic Algorithm
SP	Set Partitioning
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem
PSD	Pairwise Savings Distance
HHA	Hybrid Heuristic Approach
GAP	Generalized Assignment Problem
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
PSM	Producer Scrounger Method
CVRP	Capacitated Vehicle Routing Problem
CFRS	Cluster-First-Route-Second
RFCS	Route-First-Cluster-Second
VTPSO	Velocity Tentative Particle Swarm Optimization

Chapter I
Introduction

The Vehicle Routing Problem (VRP) is one of the most studied combinatorial optimization problems and is concerned with the optimal design of routes to be used by a fleet of vehicles to serve a set of customers. Capacitated VRP (CVRP) is the most common and basic variant of the problem which considers equal capacities for all vehicles. The most popular way of solving CVRP is splitting the task into two different phases: firstly, assigning customers under different vehicles and secondly, finding optimal route of each vehicle. This chapter introduces CVRP and basic solution methods of it. Finally, it also contains thesis objectives and organization of the thesis.

1.1 Introduction to VRP and CVRP

VRP [1-6] is a complex combinatorial optimization problem which was first introduced by Dantzig and Ramser in 1959 [1] and it has been widely studied since. The VRP can be described as the problem of designing optimal delivery or collecting routes from one or several depots to a number of geographically scattered customers, subject to side constraints. The most general form of VRP is the Capacitated VRP (CVRP) [7-12] considering equal capacities for all vehicles. The VRP plays a central role in the fields of physical distribution and logistics. The primary objective of VRP is to reduce the number of vehicles, the total travel distance or even the time spent on the road. But most of the times these objectives, are considered simultaneously, in a hierarchical fashion.

In general, CVRP consists of a depot and a set of customers with known demands. Each customer is assigned to exactly one vehicle route. Each vehicle starts from a depot and delivers the goods required, then returns to the depot. Since vehicle has capacity limit the total demand of any route must not exceed its capacity.

The CVRP is considered to be the classical version of the VRP. A formal definition of the problem is as follows. Let $G = (V, E)$ be a complete graph with a set of vertices $V = \{0,\ldots n\}$, where the vertex 0 represents the depot and the remaining ones the customers. Each edge $\{i,j \in E\}$ has a non-negative cost c_{ij} and each customer $i \in V' = V \backslash \{0\}$ has a demand d_i. Let $C = \{1,\ldots\ldots,m\}$ be the set of homogeneous vehicles with capacity Q. The CVRP consists in constructing a set up to m routes in such a way that: (i) every route starts and ends at the depot;

(ii) all the demands are accomplished; (iii) the vehicle's capacity is not exceeded; (iv) a customer is visited by only a single vehicle; (v) the sum of costs is minimized.

There is relation between CVRP and TSP: CVRP may be considered as a modification and/or update of TSP. TSP is the problem of finding a minimal length closed tour that visits all cities of a given set exactly once. It does not consider any capacity limit. On the other hand, CVRP generates several routes considering vehicle capacity and depot is the common in all the routes. Therefore, route finding of each vehicle in CVRP is a small TSP problem. Figure 1.1 demonstrates TSP and CVRP.

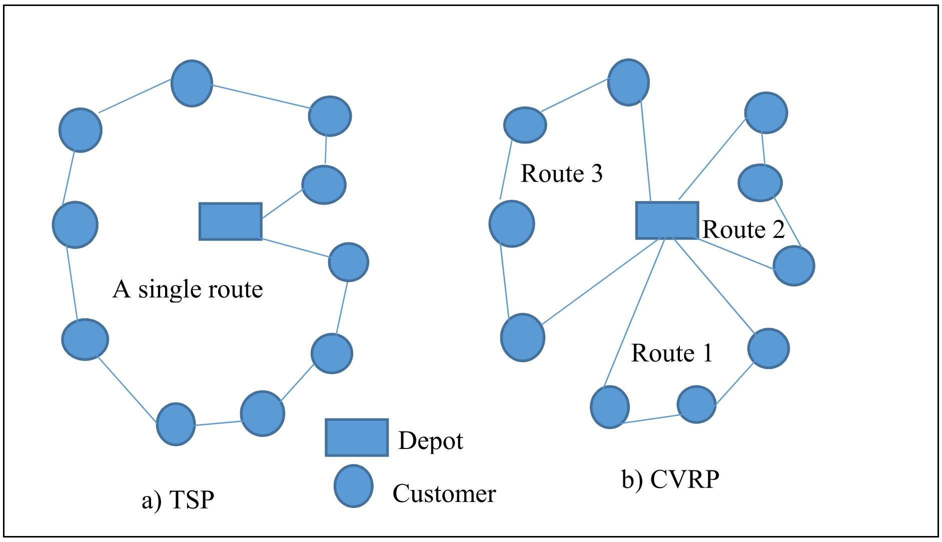


Fig. 1.1: Comparison between TSP and CVRP.

1.2 Variants of CVRP

The VRP arises naturally as a central problem in the fields of transportation, distribution, and logistics [1]. In some market sectors, transportation means a high percentage of the value added to goods. Therefore, the utilization of computerized methods for transportation often results in significant savings ranging from 5% to 20% in the total costs, as reported in [3]. In some real world VRPs there are often side constraints due to other restrictions. Some of the well-known models [3, 5, 9] are:

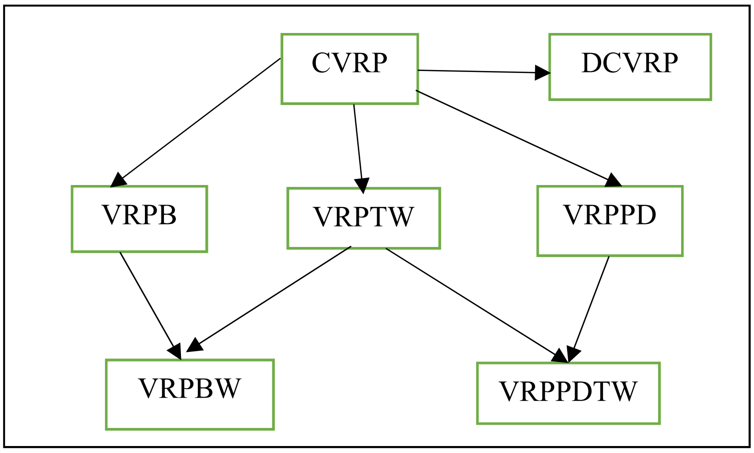


Fig 1.2: Variants of CVRP

- Capacitated Vehicle Routing Problem (CVRP): The CVRP is the simplest and the most studied problem. CVRP works with predefined demands and locations of customers to serve/delivery. The delivery for a customer cannot be split. In other words, the demand of a customer must be satisfied via only one visit. All vehicles are assumed to have the same loading capacity. They depart from a single depot at the beginning and return to the depot at the end. The service or delivery time for each customer may or may not be considered. The objective is to minimize the total traveling distance or time for all vehicles to serve all customers.
- Distance-constrained Vehicle Routing Problem (DVRP)**:** The DVRP is a variant of the CVRP. Each route of a vehicle is constrained by a maximum length of distance or time. Because of the distance constraint, the total traveling distance in each route cannot exceed the maximum prescribed length.
- Vehicle Routing Problem with Time Windows (VRPTW): The VRPTW is another variant of the CVRP. In the VRPTW, the distance constraint may or may not be considered. Each customer has a time interval, referred to as a time window. The visit of a vehicle to a customer must occur within his or her time window. In case of early arrival at a customer's location, the vehicle is allowed to wait until the beginning of the customer's time window. The time windows are defined by assuming that all vehicles start from a depot at the beginning.
- Vehicle routing problem with pickup and delivery (VRPPD): In a VRPPD, vehicles are required not only to deliver products to a set of delivery locations, but also to pick goods or wastes up at a set of pickup locations. Unlike other VRPs, products to be delivered

are not provided at the depot; rather, they must be picked up. For multiple pickups, the loading capacity of a vehicle must be considered in the problem. Time windows for the pickup and the delivery at each location may or may not be considered in the problem.

- Vehicle Routing Problem with LIFO: Similar to the VRPPD, except an additional restriction is placed on the loading of the vehicles: at any delivery location, the item being delivered must be the item most recently picked up. This scheme reduces the loading and unloading times at temporarily unload items other than the ones that should be dropped off.
- Multiple Depot Vehicle Routing Problem (MDVRP): The vendor uses many depots to supply the customers.
- Split delivery Vehicle routing problem (SDVRP): The customers may be served by different vehicles.
- Stochastic vehicle routing problem (SVRP): The demands, service time and/or travel time are random.
- Vehicle routing problem with backhauls (VRPB): A VRP in which customers can demand or return some commodities.

1.3 CVRP Solving Techniques

The CVRP is a well-known integer programming problem which falls into the category of NP-Hard problems, which means that the computational effort required to solve this problem increases exponentially with the problem size. For such problems it is often desirable to obtain approximated solutions, so that they can be found quickly enough and are sufficiently accurate for the purpose. Usually this task is accomplished by using various heuristic methods, which rely on some insight into the nature of the problem.

CVRP is a popular combinatorial optimization problem and a number of methods have been investigated for solving CVRP up to date. The methods include exact approaches, constructive methods, two-phase way etc [3, 6]. Exact approaches compute every possible solution until the best is reached. Branch-and-bound and branch-and-cut algorithms are quite popular in the category of exact approaches [3, 13-15]. Constructive methods use two main techniques for constructing VRP solutions: merging existing routes using a savings criterion, and gradually assigning vertices to vehicle routes using an insertion cost. The well-known algorithms are Clarke and Wright Savings, matching based, multi-route improvement heuristics [16]. In 2-

Phase way the CVRP is solved into two different phases: assigning customers under different vehicles and finding optimal route of each vehicle. 2-Phase ways also categorized into two groups: (i) Cluster-First Route-Second (CFRS) and (ii) Route-First-Cluster-Second (RFCS). Sweep algorithm is well studied in 2-phase way for clustering customers into groups so that customers in the same group are geographically close together and can be served by the same vehicle [10, 17-23].

Route optimization for a vehicle is simply a TSP [18] and a number of optimization methods are investigated for this purpose. Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS) with their numerous variations are used for CVRP route optimization [2, 24, 25]. The pioneer Swarm Intelligence (SI) methods Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are also found efficient for route optimization [26, 27].

1.3 Objectives of the Thesis

Among the existing CVRP solving techniques, CFRS with Sweep algorithm clustering is widely studied. The aim of the study is to investigate better CVRP solving technique in the line of CFRS. Limitation of Sweep is investigated and a variant of it proposed to achieve better clustering. On the other hand, popular TSP methods, including recently developed methods, are investigated for route optimization. To reach the goal this study will be carried out with the following specific objectives:

- Study of CVRP and existing ways to solve it.
- Investigate Sweep clustering algorithm for vehicle wise node assignment.
- Investigate a variant of Sweep algorithm owing to achieve optimal clusters.
- Study of existing optimization techniques of route generation of individual vehicle.
- Investigate SI based methods to find optimal route for individual cluster.
- Compare performance on the benchmark CVRPs to identify effective method of CVRP.

1.4 Organization of the Thesis

The main attraction of this thesis is the variant Sweep algorithm for efficient vehicle assignment and SI methods to optimize routes. The thesis has five chapters. An introduction to CVRP and its applications to solve optimization tasks has been given in Chapter I. Chapter wise overviews of rest of the thesis are as follows.

Chapter II is for literature review that includes description CVRP with its constraints and existing solution methods to solve CVRP. The chapter also identifies the legging of existing methods and gives motivation to development of new method.

Chapter III explains the proposed variant Sweep algorithm and process of route optimization in detail.

Chapter IV reports the experimental results for the benchmark problems for variant Sweep clustering and route optimizing using GA, ACO, PSM and VTPSO.

Chapter V is for the conclusions of this thesis together with the outline of future directions of research opened by this work.

Chapter II

Literature Review

CVRP is a constraint satisfaction problem in which customers are served by finite number of vehicles and try to keep total travel distance of the vehicles as minimum as possible. Various methods have been investigated to solve CVRP in last few decades and the methods may categorize as exact approaches, constructive methods, two-phase approaches etc. A two-phase approach first clusters customers to assign into different vehicles and then determines optimal route of each vehicle. Sweep algorithm is the most popular for clustering in the two-phase ways which is rigorously studied in this thesis. This chapter categorically reviews existing methods to solve CVRP. As a conclusion this chapter draws the scope of the research.

2.1 CVRP and its Constraints

CVRP is a real life constraint satisfaction problem in which customers are assigned to individual vehicles considering their capacity; and the objective of CVRP is to minimize the total traveling distance for all vehicles to serve all customers [7-12, 28, 29]. CVRP works with predefined demands and locations of customers to serve. The service/delivery for a customer cannot be split, i.e., the demand of a customer must be satisfied via only one visit. All vehicles are assumed to have the same loading capacity and they depart from a single depot at the beginning and return to the depot at the end. The service or delivery time at each customer may or may not be considered.

Figure 2.1 shows the pictorial view of CVRP: the rectangle in the center indicates depot and circles indicate customers. Every customer has its identification number and also there is a demand *(D)* with them. There are four vehicles available with capacity 10. The demand of each

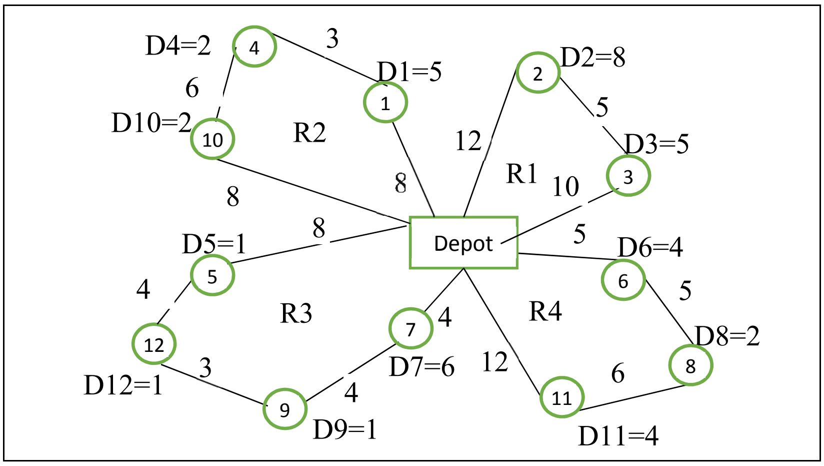


Fig. 2.1: A typical CVRP [30].

customer and the travel cost of each edge in the solution are given in the figure. A feasible route is given by *R1 = (0, 3, 2, 0)*. The cost and load of the route are 27 and 13, respectively. A complete solution of CVRP is given by *x = {R1, R2, R3, R4}.*

CVRP is important due to its various practical applications. For example, garbage collection companies need to plan the routes for collecting the garbages in the urban area; bus companies need to plan the time and the routes for buses and drivers. Other practical CVRP applications include mail delivery, street cleaning, school bus routing, routing of salespeople and maintenance units, transportation of handicapped people, heating oil distribution, parcel pick-up and delivery, and many others [9, 31].

Some studies are available to solve specific CVRP task. Xiaoyan Li [32] studied CVRP for solving the delivery problem of St. Mary's Food Bank's Distribution Center and minimized the number of trucks and total travel time while picking up all goods from 54 donors. Wen et al. [30] managed transportation of a Danish consultancy Transvision using identical vehicles to transport orders from suppliers to customers. Demiral et al. [33] constructed a method to solve School Bus Routing Problem. They founded optimal school bus routes for the Isparta Milli Piyango Anadolu High School, Isparta, Turkey. S. Amini et al. [34] proposed a method for a real-world case study of a Chlorine Capsule distribution company to the water reservoir in Tehran. Their results indicate that the algorithm can reduce the cost and time significantly. Faulin et al. [35] solved the logistic problems related to a canning company situated in Navarre, Spain. They discussed several solutions to a real case in the agribusiness sector in Navarre, Spain. In [12] the authors solved the delivery problem of Coca-Cola distribution center in Rajshahi City Corporation, Bangladesh to minimize the traveling distance and to determine an optimal distribution plan that meets all the demands

A number of constrains have to be maintained to solve CVRP. Mathematically, a CVRP is defined as

$$\text{Minimize} \sum_{i\in N}\sum_{j\in N}\sum_{v\in V} C_{ij} X_{ij}^{v} \tag{2.1}$$

$$\text{Subject to} \sum_{v\in V} y_i^v = 1 \quad \text{for} \quad i \in N \tag{2.2}$$

$$\sum_{i\in N} x_{ij}^{v} = y_j^v \ \text{for} \ j \in N \ \text{and} \ v \in V \tag{2.3}$$

$$\sum_{j \in N} x_{ij}^{v} = y_{i}^{v} \text{ for } i \in N \text{ and } v \in V \quad (2.4)$$

$$\sum_{i \in N} d_{i} y_{i}^{v} \leq Q \text{ for } v \in V \quad (2.5)$$

$$\sum_{i \in N} x_{i1}^{v} \leq 1 \text{ for } v \in V \quad (2.6)$$

$$\sum_{j \in N} x_{1j}^{v} \leq 1 \text{ for } v \in V \quad (2.7)$$

In this formulation, the objective function is expressed by Eq. (2.1) which states that the total travelling distance of all vehicles is to be minimized. Eq. (2.2) represents the constraint that each customer must be visited once by one vehicle, where $y_i^v = 1$ if vehicle v visits customer i, and 0 otherwise. It is guaranteed in Eq. (2.3) and Eq. (2.4) that each customer is visited and left with the same vehicle, where $x_{ij}^v = 1$ if vehicle v travels from customer i to customer j, and 0 otherwise. A constraint in Eq. (2.5) ensures that the total delivery demands of vehicle v do not exceed the vehicle capacity. Eq. (2.6) and Eq. (2.7) express that vehicle availability should not be exceeded.

2.2 Existing Approaches to Solve CVRP

Various methods have been investigated to solve CVRP in last few decades. Fig. 2.2 shows the common solution methods of CVRP [36]. The existing methods may categorize as exact approaches, constructive methods, two phase way etc. The following subsection briefly describes existing approaches under different categories.

2.2.1 Exact Approaches

Exact approaches compute every possible solution and best one is considered as an outcome [16]. Branch-and-Bound and Branch-and-Cut algorithms are quite popular in the category of exact approaches [3, 13-15]. The common problem of such approaches is the huge time to check all possible solutions which increases with the problem size.

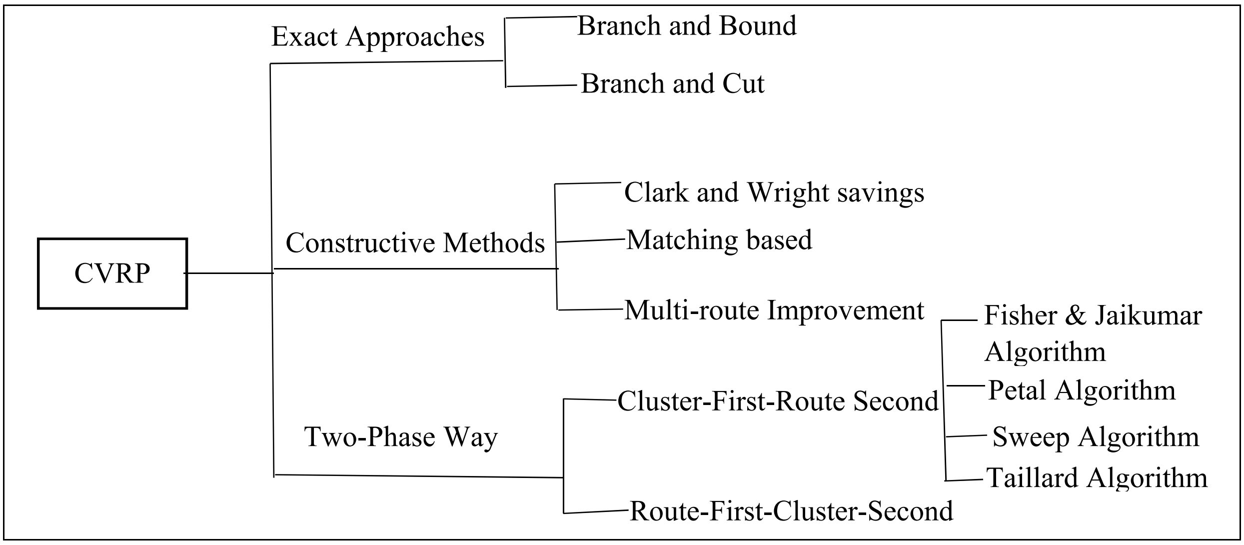


Fig 2.2: Solution methods of CVRP.

2.2.1.1 Branch and Bound Algorithm

Branch and Bound is one of the pioneer techniques for solving combinatorial problems such as CVRP [7, 15, 37, 39]. The principle idea of Branch and Bound is to divide the main problem into sub-problems (branching), and evaluate the lower and upper bounds for these sub-problems (bounding). If it is found that a sub-problem does not contain the optimal solution, it is discarded (pruning). Otherwise, the sub-problem will be further branched and bounded. When dealing with a maximization problem, an upper bound is used, and analogously when handling a minimization problem such as the CVRP, a lower bound is used [7]. Algorithm 2.1 shows the Branch and Bound algorithm.

Several studies are available for solving CVRP using Branch-and-Bound algorithms. G. Laporte et al. [37] solved CVRP using Branch and Bound algorithm. Computational tests were performed on a number of randomly generated problems. The algorithm was quite successful in solving problems involving up to 100 cities. Christofides et al. [13] solved CVRP based on spanning trees using Branch and Bound. They presented computational tests for a number of problems covering only 25 customers. Toth et al. [3] reviewed several exact algorithms based

Algorithm 2.1: Branch and Bound Algorithm

1. ***Initialization***

   Compute lower and upper bounds

   a. Set $L_1 = \Phi_{lb}(S)$ and $U_1 = \Phi_{ub}(S)$, where S is the main problem and Φ_{lb}, Φ_{ub} are the lower and upper bound functions.

   b. Terminate if $U_1 \leq L_1$.

2. ***Branching***

   Partition (split) S into two sub-problems as $S = S_1 \cup S_2$

3. ***Bounding***

   a. Compute lower and upper bounds for the sub-problems.

   b. Update lower and upper bounds on optimum solution.

      i. Update lower bound: $L_2 = \min\{\Phi_{lb}(S_1), \Phi_{lb}(S_2)\}$

      ii. Update upper bound: $U_2 = \min\{\Phi_{ub}(S_1), \Phi_{ub}(S_2)\}$

      iii. Terminate if $U_2 \leq L_2$.

   c. Split *S1* or *S2*, and repeat Step 3(a).

on the branch and bound approach proposed in the last few years for the solution of CVRP. They showed the computational results comparing the performance of different algorithms on a set of benchmark instances [38]. Dastghaibifard et al. [39] proposed a new parallel branch and bound algorithm for the solution of CVRP. They did experiments on the so-called A, B, and P benchmark CVRP instances [40] with the parallel algorithm and obtained good result.

2.2.1.2 Branch and Cut Algorithm

Branch and Cut algorithm is currently the best available exact approach for the solution of the CVRP [14]. The algorithm attempts to strengthen the Linear Programming Relaxation (LPR) of an Integer Program (IP) with new inequalities before branching a partial solution. Linear programming relaxation is the problem that arises by replacing the constraint that each variable must be 0 or 1 by a weaker constraint, that each variable belong to the interval [0,1] [41]. An integer program (IP) is a linear program in which all variables must be integers [42]. In Branch and Cut algorithm the main problem is split into multiple (usually two) versions. The new linear programs are then solved using the simplex method and the process repeats [43]. The simplex method is a method for solving problems in linear programming which tests adjacent vertices of the feasible set in sequence so that at each new vertex the objective function improves or is

Algorithm 2.2: Branch and Cut Algorithm

1. Initialization

a. Take $\overline{Z}$ = Solution of relaxed problem and Z^*= cost of best solution.

b. Divide the main problem into sub-problems.

c. Solve the problem using simplex method to obtain $\overline{Z}$.

2. Feasibility Check

a. Compare $\overline{Z}$ with the Z^*. If $\overline{Z} \geq Z^*$ update the list of sub problems and choose the next sub problem then start from Step 1, otherwise continue.

b. Force the variables that are not in the sub-problem to zero.

c. Purge ineffective constraints.

d. Generate distance and capacity constraints.

e. Generate several cuts.

3. Branching

a. Create new sub problems by branching. If the solution is integer then update Z* and continue.

b. Backup search tree.

c. Update the list of problems.

4. Terminate if the list of sub problems empty. Otherwise choose the next sub-problem and go to Step 1.

unchanged [40]. In the first step of the algorithm, non-integral solutions to LP relaxations serve as upper bounds and integral solutions serve as lower bounds [43]. Branch and Cut algorithm of Laporte, Nobert and Desrochers (1985) [15] for CVRP is described in Algorithm 2.2.

Several studies are available for solving CVRP using Branch and Cut algorithms. Augerat et al. [31] was the first to describe an exact Branch and Cut algorithm for the CVRP. Toth et al. [65] solved E-VRP benchmark problems with Branch and Cut algorithm and achieved better performance. Ralphs et al. [62] proposed a Branch and Cut algorithm and solved the A-VRP and B-VRP benchmark CVRP problems. Lysgaard et al. [44] investigated a new Branch and Cut algorithm for CVRP and solved three instances of Augerat with optimal value for the first time. Baldacci et al. [66] also developed a Branch and Cut approach CVRP and CVRP with time window. They solved six classes of instances and compared performance among the methods.

2.2.2 Constructive Methods

Constructive methods use two main heuristic techniques for solving CVRP: merging existing routes using a savings criterion, and gradually assigning vertices to vehicle routes using an insertion cost. The well-known algorithms in this category are: i) Clarke and Wright Savings Algorithm and ii) Matching based Savings Algorithm. Such heuristic approaches are straight forward but unable to identify alternate solutions.

2.2.2.1 Clarke and Wright Savings Algorithm

The Clarke and Wright savings algorithm [22, 45, 46] is one of the most known constructive methods for CVRP. It was developed on by Clarke and Wright [47]. This algorithm is based on a so-called savings concept, an estimate of the cost reduction obtained by serving two customers sequentially in the same route, rather than in two separate ones. The basic savings concept expresses the cost savings obtained by joining two routes into one route as illustrated in Fig. 2.3, where point 0 represents the depot.

Fig. 2.3: Illustration of the savings concept

Initially in Fig. 2.3(a) customers *i* and *j* are visited on separate routes. An alternative to this is to visit the two customers on the same route, for example in the sequence *i-j* as illustrated in Fig. 2.3(b). Because the transportation costs are given, the savings that result from driving the route in Fig. 2.3(b) instead of the two routes in Fig. 2.3(a) can be calculated. Denoting the transportation cost between two given points *i* and *j* by c_{ij} the total transportation cost D_a in Fig. 2.3(a) is:

$$D_a = c_{0i} + c_{i0} + c_{0j} + c_{j0} \tag{2.8}$$

Equivalently, the transportation cost in Fig. 2.3(b) is:

$$D_b = c_{oi} + c_{ij} + c_{j0} \tag{2.9}$$

By combining the two routes, savings S_{ij} is generated as

$$s_{ij} = c_{i0} + c_{0j} - c_{ij} \tag{2.10}$$

Therefore, the more the savings distance (S_{ij}) between two customers, the more distant they are from the origin and the closer they are to each other.

In Savings algorithm, at first pairwise savings distances (PSD) are calculated and all pairs of customer points are sorted in descending order of PSD values. It initiates a route with first node pair of PSD and subsequently checks other node pairs to insert in the route considering node adjacency and vehicle capacity. When a pair of nodes *i-j* is considered, if *i* matches with start and end node then *j* is considered for insertion if vehicle capacity allows. Route construction stops when all the customers are assigned to a vehicle. The algorithm has two versions: i) Series (creates one route at a time) and ii) Parallel (creates multiple route at a time).

In series approach, a route is competed and then process another one with remaining node pairs of PSD. Algorithm 2.3 shows the Series version of Clarke and Wright savings algorithm where

Algorithm 2.3: Clarke and Wright's Savings Algorithm (Series approach)

1. Initialization

a. Calculate pairwise savings distance (PSD) using Eq. (3).

b. Sort PSD in descending order of savings distance.

c. Set $k = 1$ // route index.

2. Route Generation

a. Create *k*th route R_k $\{0\text{-}n_s\text{-}n_e\text{-}0\}$ with the node pair of first value of *PSD*.

b. Process each of remaining node pair *(i,j)* of PSD for the current route R_k

If nodes *i or j* matches with n_s *or* n_e and vehicle capacity does not exceed with it

- R_k is updated inserting *i or j* with n_s *or* n_e.
- Remove the *i-j* pair from PSD.

c. If PSD is empty then Terminate; Otherwise $k = k + 1$ and go to Step 2.a

k denotes the route index. A route R_k is initiated (Step 2(a)) with the first pair of nodes of sorted PSD. Consequently checks other node pairs are checked to compete the route (Step 2(b)). If some nodes are remained then another route is constructed repeating Step 2(a) and Step 2(b).

Multiple routes are initiated in parallel version of Savings algorithm and a node pair is checked against all available routes. If a pair don't fit in the existing routes a new route is created with the pair, while the node pair is skipped in series approach. After each insertion, the partial routes are considered for merging. Two routes are combined into one if their edges match,

Algorithm 2.4: Clarke and Wright's Savings Algorithm (Parallel approach)

1. Initialization

a. Calculate pairwise savings distance (PSD) using Eq. (3).
b. Sort PSD in descending order of savings distance.
c. Create first route R_1 $\{0\text{-}n_s\text{-}n_e\text{-}0\}$ with the node pair of first value of PSD.
d. Set $K=1$ // Total existing routes

2. Route Generation

a. Take top node pair (i,j) of PSD
b. Check the node pair (i,j) with the existing routes R_k
If the nodes *(i or j)* matches with $\{n_s$ *or* $n_e\}$ and vehicle capacity does not exceed with its demand
- R_k is updated inserting *i or j* with n_s *or* n_e.
- Go to Step 3 // Merge Routes

Else

- $K = K+1$
- Create a new route R_K $\{0\text{-}n_s\text{-}n_e\text{-}0\}$ with the node pair of PSD.

c. Terminate If PSD is empty Otherwise Go to Step 2.a

3. Merge Routes

a. If first or last node of two routes matches such as *0-i-j-0* and *0-k-j-0*.
AND the total demand of the combined route satisfy vehicle capacity
- Merge the routes as *0-i-j-k-0*

b. Terminate when no more merge possible Otherwise Go to Step 3.a

combined route demand don't exceed the vehicle capacity and also the resulting route does not have any duplicate node. In parallel savings, only one pass requires through the savings pair list for route construction. Algorithm 2.4 shows the parallel version of Clarke and Wright Savings algorithm.

A number of studies are available for solving CVRPs based on Clarke and Wright savings algorithm. A. Poot [46] introduced a Savings based heuristic for a Dutch consultancy firm specialized in applied operations research. They used ten real-life data sets of four different companies and obtained suitable results. S. R. Venkatesan et al. [22] used Clarke and Wright savings method for A-VRP benchmark dataset. M. Straka [48] proposed a methodology based on Clarke and Wright savings method for optimization of transport planning of Tesco Prešov distribution center in Slovakia. The designed methodology brought increased efficiency in the planning and distribution process. T. Doyuran [49] discussed several enhancements of the

Clarke and Wright savings algorithm and proposed a new algorithm. Computational study on several benchmark problems revealed the effectiveness of the algorithm. T. Pichpibula [50] presented a new approach called the improved Clarke and Wright savings algorithm (ICW) to solve CVRP and tested for several benchmark CVRPs. It combines the Clarke and Wright savings algorithm with tournament and roulette wheel selection operators.

2.2.2.2 Matching based Savings Algorithm

Matching based Savings algorithm changes the original savings approach of replacing sequential and single-tour merging procedure by a matching-based procedure which merges multiple partial solutions in each step [3, 51, 52]. At the beginning it produces individual routes for each node and then try to merge the routes. The number of route merged at each iteration is determined by solving a matching problem, maximizing the savings obtained in the present iteration. At each iteration the savings (S_{pq}) obtained by merging routes *Rp* and *Rq* is computed as:

$$S_{pq} = C(R_p) + C(R_q) - C(R_p \cup R_q) \qquad (2.11)$$

Where $C(R_p)$ is the cost of optimum TSP tour over R_p. Algorithm 2.5 shows the Matching based Savings algorithm. Authors of [52] analyzed real container distribution problems in Port of Izmir, Turkey with the algorithm.

Algorithm 2.5: Matching based Savings Algorithm

1. ***Initialization***

   Generate routes for each individual node

2. ***Merge Route***
   a. Calculation Savings for Pairwise Routes using Eq. (2.11)
   b. Merge Routes considering vehicle capacity
   c. Stop while no merge possible otherwise go to Step 2.a

2.2.3 Two-Phase Way: Cluster-First-Route-Second Algorithm

Cluster-First-Route-Second (CFRS) is the most popular two-phase way of solving CVRP. CFRS first performs clustering of nodes with distinct measure and then determines vehicle routes for each cluster considering it as a TSP. The well-known CFRS are: i) Fishar and Jaikumar Algorithm, ii) Sweep Algorithm, iii) Petal Algorithm.

2.2.3.1 Fisher and Jaikumar Algorithm

Fisher and Jaikumar algorithm [3, 12, 53] first generates seeds and a cluster is built based on each seed. Generalized assignment problem (GAP) is used to minimize the distance within the cluster. The GAP seeks an allocation of jobs to capacitated resources at minimum total assignment cost, assuming a job cannot be split among multiple resource [54]. Finally, optimal vehicle route will be generated for each cluster. The basic idea of Fisher and Jaikumar approach can be described in terms of the following reformulation of CVRP as a nonlinear GAP.

$$\min \sum_{k} f(y_k) \tag{2.12}$$

s.t. $$\sum_{i} a_i y_{ik} \le b_k \qquad 1 \ldots\ldots\ldots k \tag{2.13}$$

$$\sum_{k} y_{ik} = \begin{cases} k, & i = 0 \\ 1, & i = 1 \;\ldots\ldots\; n \end{cases} \tag{2.14}$$

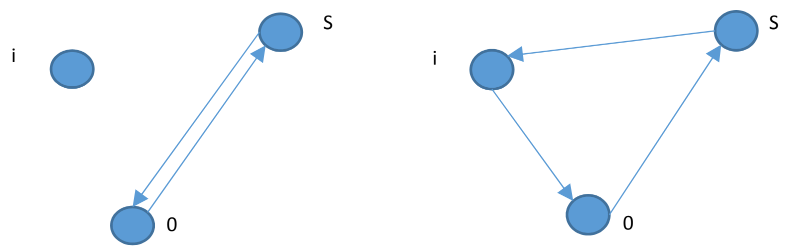
$$y_{ik} = 0 or 1 \qquad \begin{matrix} i=0\ldots\ldots\ldots\ldots n \\ k=1\ldots\ldots\ldots\ldots k \end{matrix} \tag{2.15}$$

where $f(y_k)$ is the cost of an optimal TSP tour of the customer in $N(y_k)=\{i \mid y_{ik}=1\}$. The heuristic is based on constructing a linear approximation $\sum_{i=1}^{n} d_{ik} y_{ik}$ of $f(y_k)$ and solving (10), (11), (12), (13) with (10) replaced by

$$\sum_{k=1}^{k} \sum_{i=1}^{n} d_{ik} y_{ik} \tag{2.16}$$

The solution of this GAP defines a feasible assignment of customers to vehicles. The number of vehicles *(K)* is fixed. Assignment cost is given by

$$C_{ik} = Dist(0,i) + Dist(i,S_k) \quad Dist(0,S_k) \tag{2.17}$$



a. Seed customer route   b. Visiting both seed and customer *i*

Fig. 2.4: Example of Fisher and Jaikumar algorithm

In the figures, 0 denotes the depot, S is the seed customer and i is the customer being considered for insertion. Fig. 2.4(a) shows the back and forth route of seed customer and depot. Fig. 2.4(b) shows the route if customer i is visited in the way of returning from seed customer to depot. Now the travelled distance d_S of visiting the seed customer and returning to depot is,

$$d_S = d_{0S} + d_{S0} = 2d_S \tag{2.18}$$

And if customer i is visited on the way while visiting seed, then the distance D_i,

$$d_i = d_S + d_{Si} + d_{i0} \tag{2.19}$$

Substituting Eq. (5) from Eq. (6),

$$\text{Insertion cost } C_{ik} = d_i - d_S = d_{Si} + d_{i0} - d_S \tag{2.20}$$

The algorithm is described in Algorithm 2.6.

Algorithm 2.6: Fisher and Jaikumar Algorithm

1. Seed Selection

a. Divide the nodes in angular planes by the number of vehicles: obtain cones equal to the number of vehicles.

b. Choose a seed customer from each cone that is farthest from the origin.

2. Insertion Cost Calculation

a. Calculate insertion cost for each non-seed nodes with respect to seeds using Eq. (2.20).

b. Sort the non-seed nodes descending order of distance from depot, SNS.

3. ***Assignment to Seed***

For each node from SNS

- Insert the node to a seed which gives minimum insertion cost and does not exceed vehicle capacity.

Fig 2.5 shows that the customers are divided into four cones (as number of vehicle = 4) and from each cone, the farthest node from the origin is chosen as the seed customer for that region. Then insertion is calculated for each customer with respect to their seed. In the main Fisher and Jaikumar algorithm, a geometric method based on polar angle has been proposed on the partition of the plane into K cones according to the customer weights. The seed vertices are dummy customers located along the rays bisecting the cones. Once the clusters have been determined, the TSPs are solved optimally using a constraint relaxation-based approach. Among several real life cases, Islam [12] used Fisher and Jaikumar algorithm for solving CVRP in the case study of Coca-cola deliveries in Rajshahi distribution center.

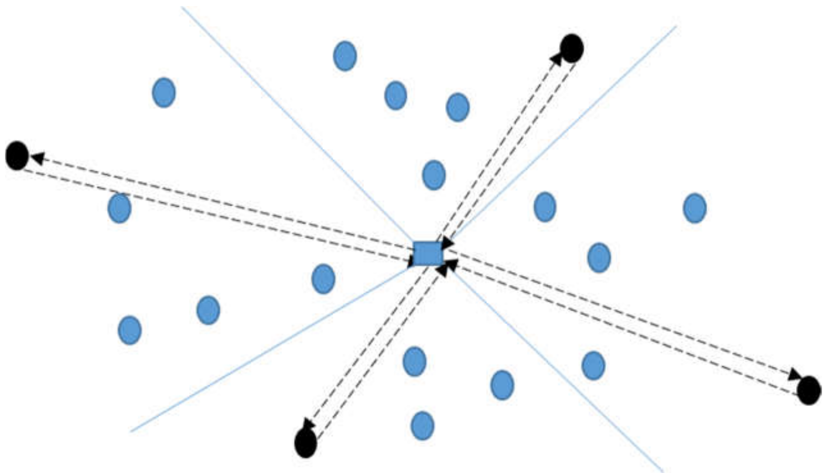


Fig. 2.5: Seed customer selection

2.2.3.2 The Sweep Algorithm

Sweep algorithm is a constrained-based heuristic which applies to planar instances of the CVRP [10, 17-23, 55]. In Sweep, feasible clusters are initially formed by rotating a ray centered at the depot. At first polar angle of each nodes are calculated using Eq. (2.21) and order the nodes according to polar angle. Then, nodes are clustered according to the polar angles. The first cluster considers the nodes from the beginning until capacity of a vehicle does not exceed. Similarly all the nodes are clustered into different vehicles. For a complete CVRP solution, vehicle route is obtained for each Sweep cluster through any TSP optimization method.

$$\theta = \tan^{-1}(y/x) \tag{2.21}$$

In the equation Θ is the angular value of a node in degree (customer location).

Figure 2.6 shows the graphical representation of cluster formation in Sweep algorithm. The dots represent the nodes and the straight lines represent the Sweep hand that moves anti-clockwise. This type of sweeping is called forward sweep. And in backward sweep, clustering

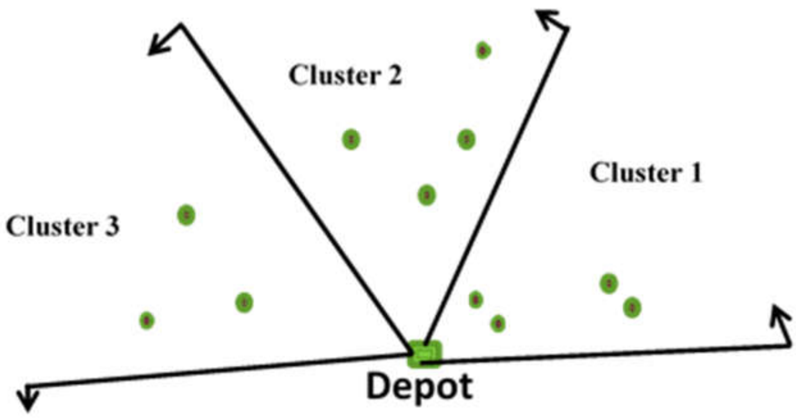


Figure 2.6: Cluster formation in Sweep algorithm

Algorithm 2.7: Sweep Algorithm
1. Initialization
a. Compute the polar coordinates of each customer using Eq. (2.21).
b. Sort the customer according to polar angles.
c. Set *C = 1*. // Cluster index
2. Clustering
a. Sweeping nodes to current cluster *C* by increasing polar angle.
b. Stop when adding the next node would exceed vehicle capacity.
c. Create a new cluster *C+1* by resuming the sweep where the last one left off.
d. Repeat Steps 2.a –2.c, until all customers have been included in a cluster.

direction is clockwise which means though clustering starts from 0^0, then it advances algorithm from 360^0 to 0^0. The Sweep algorithm is described in Algorithm 2.7.

A large number of Sweep based studies are available to solve CVRP. The capability of Sweep in solving the CVRP for public transport was investigated in [18]. Author compared the performance of routes obtained by Sweep with that of current routes. The result shows that Sweep is capable of solving vehicle routing problem for public transport under certain constraints. N. Suthikarnnarunai [19] routing problem of University of The Thai Chamber of Commerce in Bangkok using Sweep algorithm with 2-opt. A. Boonkleaw [20] proposed a Sweep algorithm to solve morning newspaper delivery problem of Bangkok, Thailand which aims to improve delivery time. Venkatesan et al. [22] solved Augerat benchmark CVRPs using Sweep algorithm.

A number of studies also incorporated different techniques to improve Sweep clustering. K. Shin [8] introduced cluster adjustment approach in Sweep and route generated with Lin-Kernighan heuristic method. Na [23] et al. introduced nearest neighbor approach in Sweep and route optimized with 2-opt edge exchange method. An additional operation with Sweep increases the computation cost in these methods.

Several hybrid studies are available that combine Sweep with other methods. M. Yousefikhoshbakht [21] proposed a hybrid algorithm combining Ant Colony System, Sweep algorithm and 3-opt local search was for solving CVRP. In [56] CVRP has been solved using Sweep algorithm jointly with Clark and Wright savings algorithm to find out best route between warehouse and distribution centers on Chen's benchmark problem. Recently, Aziz et al. [10] proposed a hybrid algorithm of Sweep algorithm and nearest neighbor algorithm for CVRP. The method tested on Augerat's Euclidean benchmark dataset and also solved the dairy products delivery problem of Tiba Company for Trade and Distribution in Egypt.

2.2.3.3 The Petal Algorithm

Petal algorithm uses Sweep concepts to create initial tours in four geometric regions. Then it competes the initial routes based on insertion cost, the additional cost incur in a tour with the node. Finally, it generates additional tours based on Nearest-Neighbor concepts with the remaining nodes [3, 57, 58]. Algorithm 2.8 shows the Petal algorithm.

Algorithm 2.8: Petal Algorithm

1. Initial Tour Creation
Create initial tours in four Geometric regions using Sweep considering vehicle capacity.
// Maximum number of initial tour is four

2. Complete the Initial Tours
For each or remaining node
If vehicle capacity does not exceed with its insertion
- Insert it in different positions of each initial tour and compute insertion cost
- Assign to the initial tour for smallest insertion cost

***3.* Tour Creation with Remaining Nodes**
Create additional tours with Nearest-Neighbor method considering vehicle capacity.

Ryan et al. [58] investigated an efficient shortest path technique for producing the optimal petal solution solving CVRP. They extended the definition of a petal route and showed that the optimal generalized petal solution can be produced efficiently by multiple applications of a shortest path algorithm. J. Renaud et al. [57] proposed an improved version of the original petal algorithm that generates a set of good feasible routes and selects some of these routes using a set partitioning problem [59]. C. Hjorring [60] presented a genetic algorithm based heuristic for CVRP which uses a specially designed crossover that combines the cyclic orders of two solutions to form a new cyclic order. The petal method was used to transform the cyclic order into a CVRP solution.

2.2.4 Two Phase Way: Route-First-Cluster-Second Algorithm

Route-First-Cluster-Second (RFCS) method [61] constructs a giant TSP tour with all the nodes in first phase disregarding other constraints, and decompose this tour into feasible vehicle routes in second phase. This idea applies to problems with a free number of vehicles. It was first put forward by Beasley who observed that the second phase problem is a standard shortest path problem on an acyclic graph. Algorithm 2.9 shows the RFCS steps and Fig. 2.9 shows a demonstration of it.

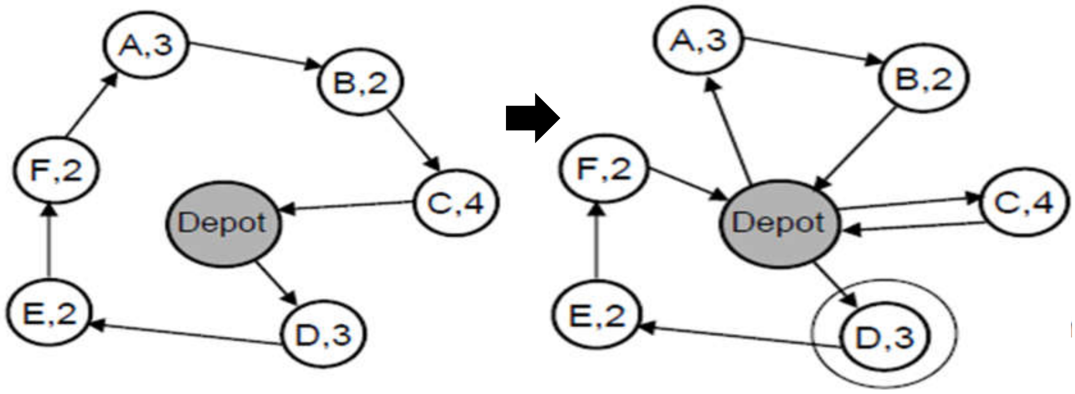


Fig. 2.7: Demonstration of Route-First and Cluster-Second method.

Algorithm 2.9: Route-First-Cluster-Second Algorithm

1. Routing

Solve a single TSP problem with the nodes relaxing the vehicle capacity constraint.

2. Clustering

Cut the TSP solution into routes that satisfy the vehicle capacity.

2.3 Scope of the Research

The CFRS is most popular way of solving CVRP: firstly, assigning customers under different vehicles and secondly, finding optimal route of each vehicle. Among various solution methods of CVRP Sweep algorithm is well studied for clustering customers in which cluster formation starts from 0^0 and consequently advances toward 360^0 to consider all the nodes. Such rigid starting is identified that total cluster formation may exceed total number of available vehicles for some instances. It is worth mentionable that starting from different angles may give different clusters and explores chance to get better CVRP solution after route optimization. Therefore, a variant of Sweep algorithm considering starting angle as a user defined parameter might give better node assignments to individual vehicles

Route generation is aimed to link all nodes in every cluster starting from and ending to the same depot. Route generation of individual vehicle is a TSP optimization task and any TSP optimization method is useful. Particle swarm optimization (PSO) has been used in some studies to find optimum vehicle route. Other prominent methods such as Genetic Algorithm and Ant Colony Optimization [26, 27] as well as recently proposed Producer-Scrounger Method [63] and Velocity Tentative PSO (VTPSO) [64] might be interesting for route optimization and may achieve better outcome.

Chapter III

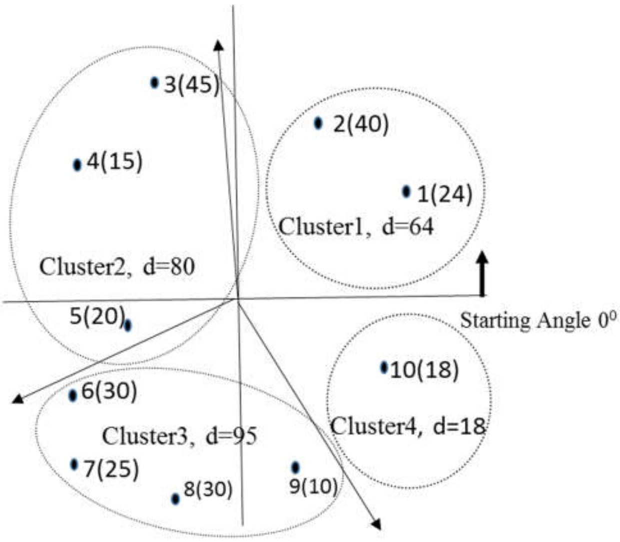
Solving CVRP Using Variant Sweep and Swarm Intelligence

The aim of this study is to identify the effective Capacitated Vehicle Routing Problem (CVRP) solving method considering Sweep algorithm to vehicle wise cluster the nodes and Swarm Intelligence (SI) based methods to optimize route of each vehicle. A variant version of Sweep is considered in this study for better outcome. Route optimization is a traveling salesman problem (TSP); and therefore, prominent TSP solving methods including Genetic Algorithm, Ant Colony Optimization and Particle Swarm Optimization are considered in this study. This chapter first describes the proposed variant Sweep for clustering nodes and then briefly discusses the route optimization methods.

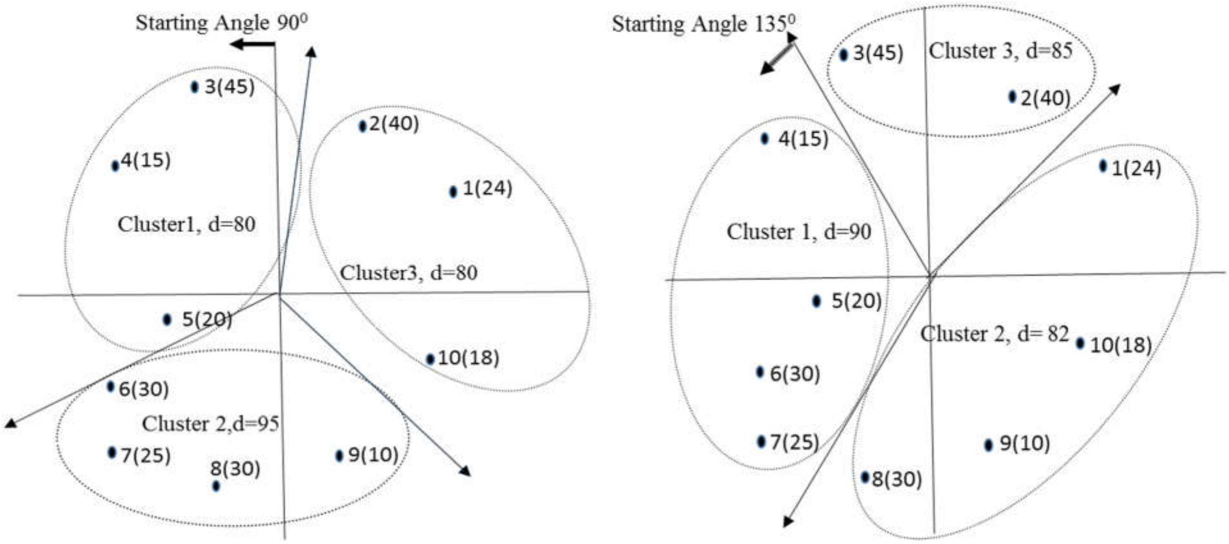
3.1 Variant Sweep Clustering

It is already described in the previous section that standard Sweep considers polar angle of nodes and capacity of vehicle. In general, standard Sweep considers depot located at (0, 0) co-ordinate in two dimensional plane. It first calculates polar angle of each individual node and order the nodes according to polar angle. Finally, cluster formation starts from 0^0 and consequently advance toward 360^0 to assign all the nodes under different vehicles considering vehicle capacity [19, 22]. Problem with such rigid starting from 0^0 is identified that total clusters formation may exceeds total number of available vehicles for some instances. It is worth mentionable that cluster formation may differ for different starting angles and explores chance to get better CVRP solution after route optimization.

Figure 3.1 demonstrates the inadequacy with standard Sweep and its improvement way for a sample CVRP. The CVRP consists with 10 nodes with different demands around the depot and the total demand of the nodes 157 will be served vehicles having capacity 100. Fig. 3.1(a) shows the cluster formation with standard Sweep starting from 0^0: Cluster 1 covers demand 64 with two nodes, Cluster 2 covers demand 80 with two nodes, Cluster 3 covers demand 95 with four nodes; and remaining demand 18 is assigned in Cluster 4.Therefore, required number of vehicles in standard Sweep is 4. But three vehicles (total capacity 100*3=300) might enough to serve all the nodes heaving total demand 157. Fig. 3.1 (b) shows cluster formation with Sweep technique but starting from 90^0 in which all the nodes are assigned into three clusters each one demand is below vehicle capacity: Cluster 1 covers demand 80 with three nodes,



(a) Clustering of nodes through standard Sweep with starting angle 0^0.



(b) Clustering of nodes through variant Sweep with starting angle 90^0 and 135^0.

Fig. 3.1: Clustering of nodes with standard and variant sweep algorithms.

Cluster 2 covers demand 95 with four nodes, Cluster 3 covers remaining three nodes with demand 82. Three clusters also found sufficient to cover all the nodes for starting angle 135^0. It is obvious that total CVRP cost for three vehicles will be less than the cost for four vehicles. Therefore, this study considers the starting angle of cluster formation as user defined parameter and the method called variant Sweep.

Algorithm 3.1 shows the steps of proposed variant Sweep algorithm. First three steps of the initialization section are same as standard Sweep: update nodes' coordinates considering depot

Algorithm 3.1: Variant Sweep Algorithm

1. Initialization

a. Update coordinates of the nodes considering depot as (0, 0) in the two-dimensional plane.
b. Compute the polar angle of each node.
c. Order the nodes according to polar angle, *ONL*
d. Select starting angle of cluster formation, Θ_s
e. Cluster *C*=1

2. Clustering

a. Identify position of Θ_s in *ONL*.
b. Sweeping nodes to current cluster *C* by increasing polar angle.
c. Stop when adding the next node would exceed vehicle capacity.
d. Create a new cluster *C+1* by resuming the sweep where the last one left off.
e. Repeat **Steps 2.b –2.d**, until all customers have been included in a cluster.

Outcome

All the nodes are assigned into total *C* clusters.

as location as (0,0), compute polar angle of each node and order the nodes according to polar angle to a list *ONL*.

Cluster formation starts in variant Sweep from the defined angle Θ_s and nodes are assigned into different clusters considering vehicle capacity. First the method identify the position of Θ_s in *ONL* (Step 1). As like standard Sweep, variant method assigns nodes into a cluster while vehicle capacity does not exceed (Steps 2 and 3) otherwise new cluster forms for unassigned nodes (Step 4). Since the variant Sweep may starts any location of *ONL,* Step 5 transforms node assignment from bottom of *ONL* to the beginning of *ONL*. It is notable that for $\Theta_s = 0^0$ the proposed method will be standard Sweep and Step 5 will not be executed.

3.1.1 Selection of Cluster Formation Starting Angle (*Θs*)

It is already explained that appropriate starting angle for cluster formation is an important matters in the proposed variant Sweep algorithm. It is possible to check the proposed method with fixed different angles. But such trial and check method is required to set for every individual method. Therefore as alternative, a heuristic method is investigated in this study which aim is to identify the appropriate cluster formation starting angle (*Θs*) for a given problem.

The proposed heuristic approach consider angle difference of consecutive nodes in *ONL* and distance between the nodes and distances from the depot. The approach first calculates preference value ($p\Theta$) of each consecutive nodes and maximum $p\Theta$ is considered as the outcome of starting angle (Θs). Suppose the depot and other two consecutive nodes are D, N1 and N2, respectively. Polar angles of the nodes are $\Theta 1$ and $\Theta 2$. The distances of the nodes from the depot is *dN1* and *dN2*; and distance between the nodes is *dN12*. Fig. 3.2 shows the

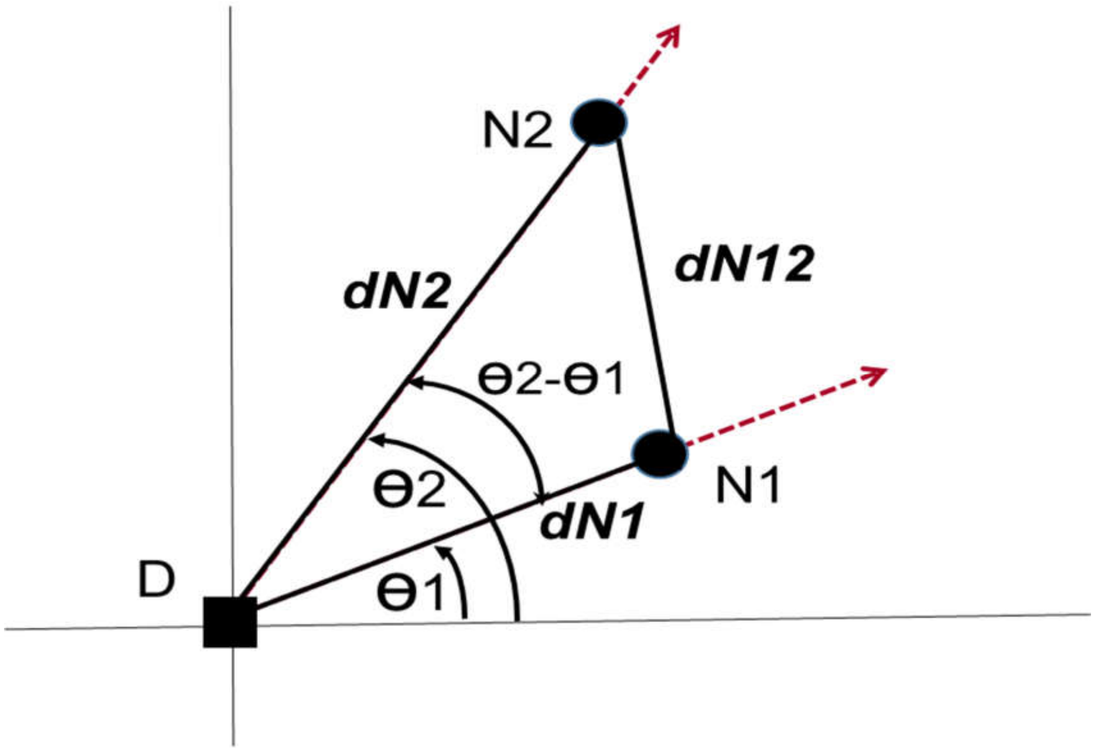


Fig. 3.2: Demonstration of cluster formation start angle selection.

graphical representation of the matter for better understanding. Preference value ($p\Theta$) for the starting angle between the nodes N1 and N2 means to place the nodes in two different clusters and is calculated using Eq. (3.1).

$$p\Theta = \alpha * (\Theta 2 - \Theta 1) + \beta * \{dN12 + Min(dN1, dN2)\} \tag{3.1}$$

In the equation, α and β are the arbitrary constants to emphasis angle difference and node distances, respectively. According to first part of Eq. (3.1), the preference value increases with angular difference of the nodes (i.e., $\Theta 2 - \Theta 1$). The second part of the equation is minimum distance to travel the two nodes from depot. If both the nodes are far from the depot as well as distance between them are large then the outcome will be large. On the other hand, $p\Theta$ value will be low even larger angle difference when both the nodes are closed to depot. After calculating the $p\Theta$ values for all the consecutive nodes, the maximum value is considered as the starting angle. If $p\Theta$ value for nodes N1 and N2 is found maximum then cluster formation will be start from N2 for anti-clock wise cluster formation.

3.2 Optimal Vehicle Route Generation

In solving CVRP, optimal route generation of each individual vehicle is a crucial part while any clustering (e.g., standard Sweep or variant Sweep) method is used to cluster nodes. In general, a clustering method divides total CVRP nodes into vehicle number of clusters. The aim of route generation is the optimal path finding of each vehicle starting from the depot and returning to depot after serving all its assigned nodes. Therefore, route generation of individual vehicle is simply a small sized TSP problem considering the depot as a common city point and any TSP optimization method may use for this purpose. To generate route for a vehicle, a TSP cost matrix considering nodes for a particular vehicle is prepared and then a TSP optimization is employed to work with the cost matrix as an independent TSP. More specifically, in sample case of Fig. 3.1, Cluster 1 belongs nodes 4, 5, 6 and 7 for $\Theta_s = 135^0$ and therefore algorithm will prepare TSP cost matrix of five cities including depot as a TSP city. Algorithm 3.2 shows the steps of vehicle route generation of individual vehicles and provide CVRP solution.

In this study, three prominent SI based methods investigated for route optimization. Genetic Algorithm (GA) also considered along with SI methods for optimization as it is a prominent and pioneer optimization method. Among the SI methods, Ant Colony Optimization (ACO) is the well-known prominent method for TSP, and Producer-Scrounger Method (PSM) and Velocity Tentative Particle Swarm Optimization (VTPSO) are two very recent well performed

Algorithm 3.2: Vehicle Route Generation Steps

1. Input

Vehicle wise nodes from variant Sweep clustering with Algorithm 3.1.

2. Route Generation for Each Vehicle

a. Include depot as a node in the cluster.
b. Prepare a TSP cost matrix with the nodes of the cluster.
c. Employ TSP optimization method to generate optimal route for the vehicle.

Outcome

CVRP solution with optimal routes of all the vehicles.

methods for TSP. Brief descriptions to optimize route of individual vehicles are explained in the following subsections.

3.2.1 Genetic Algorithm (GA)

GA is inspired by biological systems' fitness improvement through evolution and is the pioneer and widely used to solve many scientific and engineering problems. Common features of GA are: populations of chromosomes (i.e., solutions), selection according to fitness, crossover to produce new offspring, and random mutation of new offspring.

Selection: Rank based selection is common in GA [24]. It first ranks the population according to fitness of solutions. Selection of solutions for next generation is perform considering the ranks of the solutions.

Crossover: It is a process in which two chosen chromosomes combine their genetic materials to produce a new offspring which possesses both their characteristics. Two strings are picked from the mating pool at random to crossover [24]. Among several crossover techniques, only Enhanced Edge Recombination (EER) method is used to solve TSP. In EER, an adjacency table [24] (called *Edge Table*) is prepared that lists links *into* and *out of* a city found in the two parent sequences. Element of a sequence with a *common edge* is marked as inverting sign to emphasis in selection. The description of EER is available in [24].

Mutation: Mutation [24] is the process by which offsprings are generated with a single parent. Position swap of two randomly selected nodes is the common way of mutation operation for TSP.

Elitism: Elitism is a method which copies the best chromosome to the new offspring population before crossover and mutation. When creating a new population by crossover or mutation, the best chromosome might be lost. Elitism keeps the best solutions to a stack and helps to improve performance of GA.

3.2.2 Ant Colony Optimization (ACO)

ACO is inspired from ants' foraging behavior and is the prominent method for solving TSP. ACO is the first algorithm aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. It considers population size as the number cities in a given problem and starts placing different ants in different cities. A particular ant considers next city to visit based on the visibility heuristic (i.e., inverse of distance) and intensity of the pheromone on the path. After the completion of a tour, each ant lays some pheromone on the path. Before pheromone deposit, pheromone evaporation of real ant is adopted by reducing pheromone of all the links by a fixed percentage. This behavior allows the artificial ants to forget bad choices made in the past. To check whether a city has been visited or not, a tabu list is maintained which is a set of all cities that are to be visited [26]. Finally, all the ants follow the same route after certain iteration. The detail description of ACO available in [26]. If an ant in city *i,* the probability to go city *j* can be calculated by the following equation and parameters:

$$P_{i,j}^{k}(t) = \frac{[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}]^{\beta}}{\sum_{i\epsilon J_i^k}[\tau_{i,j}(t)]^{\alpha}[\eta_{i,j}]^{\beta}}, \tag{3.2}$$

J_i^k is the set of cities the ant still has to visit.

$\eta_{i,j} = \frac{1}{d_{i,j}}$ is the reciprocal of the distance from *i* to *j*.

$\tau_{i,j}$ is the amount of pheromone on the arc from *i* to *j*.

α is the importance of the intensity in the probabilistic transition.

β is the importance of the visibility of the trail segment.

After the completion of a tour, each ant lays some pheromone on the path. The pheromone is updated by the following equations.

$$\forall(i,j)\ \tau_{ij}(t+1) = (1 \quad \rho).\tau_{ij} + \sum_{k=1}^{m} \ \tau_{ij}^{k}(t) \tag{3.3}$$

$$\tau_{ij}^{k} = \begin{cases} \frac{1}{L_k} & if\ ant\ k\ uses\ edge\ i,j \\ 0 & ot\ \ erwise \end{cases} \tag{3.4}$$

ρ is the trail persistence or evaporation rate.

3.2.3 Producer Scrounger Method (PSM)

PSM [63] is a new technique to solve TSP inspiring from the animal group living behavior. It models roles and interactions of three types of animal group members: producer, scrounger and dispersed. PSM considers a producer having the best tour, few dispersed members having worse tours and scroungers. In each iteration, the producer scans for better tour, scroungers explore new tours while moving toward producer's tour; and dispersed members randomly checks new tours. For producer's scanning, PSM randomly selects a city from the producer's tour and rearranges its connection with several near cities for better tours. Swap operator and swap sequence based operation is employed in PSM to update a scrounger towards the producer. Finally, producer is considered as the solution of a given problem. The description of this method is available in [63].

Producer Scanning for Better Tour:

Producer checks (i.e., scans) several alternative tours based on current tour. The producer at first randomly selects a city (e.g., *C*) in its tour. Then select some nearest cities from *C* according to distance. Suppose one of the nearest cities is N_1. Now the producer will create connection between these two cities *C* and N_1. There are four alternative ways to connect the two selected cities. Removing *C* from its current location and placing before and after N_1 will provide two new tours. Similarly, removing N_1 from its current location and placing before and after *C* will provide other two new tours. After scanning with several cities, the producer will conceive the best scanned tour if it is better than the current tour. Number of nearest cities to check is defined as a parameter.

Scrounging to follow the Producer:

Scroungers search for opportunities to join the resources found by the producer. To solve TSP, a scrounger tries to explore better tour between tours of the scrounger itself and the producer. Swap operator (SO) and swap sequence (SS) based operation is employed for scrounging. A SO indicates two cities in a tour those positions will be swapped. Suppose, a TSP problem has ten cities and a solution is *1-2-3-6-4-5-7-8-9-10*. A *SO(4,6)* gives the new solution S'.

$$\begin{aligned} S' &= S + SO(4,6) \\ &= (1 \quad 2 \quad 3 \quad 6 \quad 4 \quad 5 \quad 7 \quad 8 \quad 9 \quad 10) + SO(4,6) \\ &= 1 \quad 2 \quad 3 \quad 5 \quad 4 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \end{aligned} \tag{3.5}$$

Here '+' means to apply SO(s) on the solution.

A swap sequence is made up of one or more swap operators.

$$SS = (SO_1, SO_2, SO_3, \ldots SO_n), \tag{3.6}$$

where $SO_1, SO_2, SO_3, \ldots SO_n$ are the swap operators. Implementation of a SS means apply all the SOs on the solution in order. The order of SOs in a SS is important [10] because implication of same SOs in different order may give different solutions from the original solution. It is notable that different SSs acting on a solution may produce the same new solution. Moreover, if applying swap sequence SS on tour *A* gives tour *B* (i.e., $B = A + SS$) then

$$SS = B \quad A. \tag{3.7}$$

To move a scrounger towards the producer, first SS is calculated using Eq. (3.7) from tours of the scrounger and the producer; and then a portion of the SS is applied on the scrounger. Implication of a portion of SS with several SOs rather than entire SS (with all SOs) ensures the scrounger to explore a new tour towards the producer. For simplicity, SS portion (i.e., number of SOs from the beginning) is considered picking a random number between 1 and total SOs in the calculated SS. Such random selection of SS portion might help to explore different tours by different scroungers as well as increase diversity in the population. A scrounger will be producer in the next iteration if it finds a better tour than the current producer and other scroungers.

3.2.4 Velocity Tentative Particle Swarm Optimization (VTPSO)

VTPSO [64] is the most recent SI based method to solve TSP extending Particle Swarm Optimization (PSO). In PSO, every particle represents a tour and changes its tour at every iteration with velocity calculated considering the best tour encountered before by itself (called as particle best) and the best tour encountered by the swarm (called as global best). SS based operation like PSM is considered for velocity calculation. In traditional PSO, the new tour of TSP is considered after applying all the SOs of a SS and no intermediate measure is considered. On the other hand, VTPSO considers the calculated velocity SS as a tentative velocity and conceives a measure called partial search (PS) to apply calculated SS to update particle's position (i.e., TSP tour).

VTPSO calculates velocity SS as like other PSO based methods. At each iteration step, VTPSO calculates velocity SS using Eq. (3.8) considering (i) last applied velocity ($v^{(t-1)}$), (ii) previous best solution of the particle (P_i) and (iii) global best solution of the swarm (G).

$$V_i^{(t)} = V_i^{(t-1)} \otimes \alpha\left(P_i \quad X_i^{(t-1)}\right) \otimes \beta\left(G \quad X_i^{(t-1)}\right) \alpha, \beta \epsilon [1,0] \tag{3.8}$$

The outcome of Eq. (3.8) is swap sequence where the negative ('-') sign operation is performed as of Eq. (3.7). VTPSO does not apply calculated velocity SS on a particle to get its new position like traditional methods. But through PS technique it measures performance of tours applying SOs of the calculated SS one after another, and the final velocity is considered for which it gives better tour. Therefore, PS technique explores the option of getting better tour considering the intermediate tours with a SS applying its SOs one by one.

Suppose $V_i^{(t)} = SO_1, SO_2, SO_3, \ldots SO_n$ then in PS

$X_i^{1(t)} = X_i^{(t-1)} + SO_1$

$$X_i^{2(t)} = X_i^{1(t)} + SO_2 = X_i^{(t-1)} + SO_1 + SO_2$$

……………………………………

$$X_i^{n(t)} = X_i^{n-1(t)} + SO_n$$

In the above cases $X_i^{1(t)}, X_i^{2(t)}, \dots\dots, X_i^{n(t)}$ are the tentative intermediate tours; and the final tour $X_i^{(t)}$ in PS is the tentative tour having the minimum tour cost.

$$X_i^{(t)} = X_i^{j(t)}, \tag{3.9}$$

where $X_i^{j(t)}$ provides the minimum tour cost among $X_i^{1(t)}, X_i^{2(t)}, \dots X_i^{j(t)} \dots X_i^{n(t)}$. Finally, the velocity considered as $V_i^{(t)} = SO_1, SO_2, SO_3, \dots SO_j \quad 1 < j \leq n$.

The final velocity may also get from new and previous positions of the particle using the equation

$$V_i^{(t)} = X_i^{(t)} \quad X_i^{(t-1)}. \tag{3.10}$$

VTPSO calculates fitness of every new position (X_i) of a particle and compares to its previous best P_i. If X_i is found better than P_i then VTPSO applies Self-Tentative operation on (X_i) owing to improve it furthermore.

Chapter IV
Experimental Studies

This chapter experimentally investigates the efficacy of proposed Variant Sweep algorithm to cluster customers and selected optimization methods for route generation. A set of benchmark problems were chosen as a test bed and performance evaluated different experimental settings. Finally, an experimental analysis has been given for better understanding of the way of performance improvement in proposed method for solving CVRP.

4.1 Bench Mark Data and General Experimental Methodology

In this study, total 51 benchmark CVRPs from two different sets of Augerat benchmark problems [16] of A-VRP and P-VRP have been considered in this study. In A-VRP, number of customer varies from 32 to 80, total demand varies from 407 to 932, number of vehicle varies from 5 to 10 and capacity of individual is 100 for all the problems. For example, A-n32-k5 has 32 customers and 5 vehicles. On the other hand, in P-VRP number of customer varies from 16 to 101, total demand varies from 246 to 22500 and vehicle capacity varies from 35 to 3000. Table 4.1 and Table 4.2 show the brief description of the A-VRP and P-VRP benchmark problems, respectively. A numeric value in the problem name presents the number of nodes and vehicles. The detailed description of the problems are available in provider's website [16]. According to Table 4.1 and Table 4.2, the selected benchmark problems belong large varieties in number of nodes, vehicles and demands; and therefore, provides a diverse test bed.

Benchmark problems are required to use in the experiments. A customer is represented as a co-ordinate in a problem. Coordinates are updated considering depot as [0, 0] for easy calculation. Distance matrix is prepared using the coordinates. Polar angle of each customer is calculated using Eq. (2.21) for angle based sweep operation. Standard Sweep (i.e., $\Theta_s = 0^0$) does not have any parameter to set and it starts cluster formation from 0^0 (i.e., $\Theta_s = 0^0$). In variant Sweep, the values of α and β were set to 0.6 and 0.2, respectively and found effective for most of the problems. In few other problems α and β values are tuned between 0.2 and 0.6. Both anti clock and clock wise sweep operations are considered in both standard and variant Sweep algorithm.

Table 4.1: Description of A-VRPs benchmark problems for CVRP.

Sl.	Problem Name	Total Nodes	Number of Vehicle	Individual Vehicle Capacity	Total Demand
1	A-n32-k5	32	5	100	410
2	A-n33-k5	33	5	100	446
3	A-n33-k6	33	6	100	541
4	A-n34-k5	34	5	100	460
5	A-n36-k5	36	5	100	442
6	A-n37-k5	37	5	100	407
7	A-n37-k6	37	6	100	570
8	A-n38-k5	38	5	100	481
9	A-n39-k5	39	5	100	475
10	A-n39-k6	39	6	100	526
11	A-n44-k6	44	6	100	570
12	A-n45-k6	45	6	100	593
13	A-n45-k7	45	7	100	634
14	A-n46-k7	46	7	100	603
15	A-n48-k7	48	7	100	626
16	A-n53-k7	53	7	100	664
17	A-n54-k7	54	7	100	669
18	A-n55-k9	55	9	100	839
19	A-n60-k9	60	9	100	829
20	A-n61-k9	61	9	100	885
21	A-n62-k8	62	8	100	733
22	A-n63-k9	63	9	100	873
23	A-n63-k10	63	10	100	932
24	A-n64-k9	64	9	100	848
25	A-n65-k9	65	9	100	877
26	A-n69-k9	69	9	100	845
27	A-n80-k10	80	10	100	942

In this study, prominent TSP optimization methods i.e., GA, ACO, PSM and VTPSO, are applied to generate optimal route of cluster-wise individual vehicles to get final CVRP solution. A fair experimental setting was maintained for each of optimization method for better outcome in route optimization. In ACO, *alpha* and *beta* were set to 1 and 3, respectively. On the other hand, the *RNC* (rate of near cities consideration) for producer scanning in PSM was set to 0.1. The algorithms are implemented on Visual C++ of Visual Studio 2013. The experiments have been done on a PC (Intel Core i5-3470 CPU @ 3.20 GHz CPU, 4GB RAM) with Windows 7 OS.

Table 4.2: Description of P-VRP's benchmark problems for CVRP.

Sl.	Problem Name	Total Nodes	Number of Vehicle	Individual Vehicle Capacity	Total Demand
1	P-n16-k8	16	8	35	246
2	P-n19-k2	19	2	160	310
3	P-n20-k2	20	2	160	310
4	P-n21-k2	21	2	160	298
5	P-n22-k2	22	2	160	308
6	P-n22-k8	22	8	3000	22500
7	P-n23-k8	23	8	40	313
8	P-n40-k5	40	5	140	618
9	P-n45-k5	45	5	150	692
10	P-n50-k7	50	7	150	951
11	P-n50-k8	50	8	120	951
12	P-n50-k10	50	10	100	951
13	P-n51-k10	51	10	80	777
14	P-n55-k7	55	7	170	1042
15	P-n55-k8	55	8	160	1042
16	P-n55-k10	55	10	115	1042
17	P-n55-k15	55	15	70	1042
18	P-n60-k10	60	10	120	1134
19	P-n60-k15	60	15	80	1134
20	P-n65-k10	65	10	130	1219
21	P-n70-k10	70	10	135	1313
22	P-n76-k4	76	4	350	1364
23	P-n76-k5	76	5	280	1364
24	P-n101-k4	101	4	400	1458

4.2 Detailed Experimental Observation on Selected Problems

This section presents detailed results for two selected problems A-n53-k7 and P-n65-k10. The population size of GA, PSM and VTPSO was 100; whereas, number of ants in ACO was equal to the number of nodes assigned to a vehicle as it desired. The number of iteration was set at 200 for the algorithms. For better understanding, experiments conducted for four fixed starting angles ($\Theta_s = 0^0$, 90^0, 180^0 and 270^0) along with adaptively selected angle. It is notable that $\Theta_s = 0^0$ represents standard Sweep.

Table 4.3 shows the total clusters for different starting angles (Θ_s) in variant Sweep and optimized route cost with different methods for A-n53-k7 problem. The problem has 53 nodes and total 664 demand to be served with seven vehicle having capacity 100. From the table it is observed that total number of clusters for $\Theta_s = 0^0$ (i.e., in standard Sweep) is 8 that is more than

Table 4.3: Clusters for different starting angle (Θ_s) in variant Sweep and CVRP cost after route optimizing using GA, ACO, PSM and VTPSO for A-n53-k7 problem.

Θs	Clusters	CVRP Cost Before Route Optimizing	CVRP cost after optimizing with			
			GA	ACO	PSM	VTPSO
0^0	8	1604	1175	1211	1174	1174
90^0	7	1654	1125	1160	1109	1109
180^0	7	1504	**1091**	**1131**	**1090**	**1090**
270^0	8	1775	1171	1196	1171	1171
220.6^0*	7	1504	**1091**	**1131**	**1090**	**1090**

* *Angle selected adaptively through proposed heuristic approach.*

available vehicles. Total clusters are also 8 for $\Theta_{s=}$ 270^0. On the other hand, number of clusters equal to total vehicles (i.e., 7) for $\Theta_{s=}$ 90^0 and 180^0. It is also remarkable that CVRP cost (i.e., total travel distance) for 7 clusters is lower than the cases of 8 clusters after route optimization. It is interesting from the table that total clusters are also 7 for adaptively selected angle 220.6^0. The best CVRP cost for an algorithm among different Θ_s is marked as bold-faced type. For the problem the best CVRP cost achieved after optimizing for with GA, ACO, PSM and VTPSO are 1091, 1131, 1190 and 1090, respectively. The best values are found for adaptively selected $\Theta_s = 220.6^0$ and fixed $\Theta_s=180^0$.

The results of Table 4.3 for problem A-n53-k7 clearly indicate that proposed heuristic approach is able to select appropriate starting angle in variant Sweep and hence better CVRP solution. It is also observed that the CVRP solutions are same for both $\Theta_s = 220.6^0$ and $\Theta_s=180^0$. For the problem, the angle difference between the angle wise consecutive nodes 33 (at 146.31^0) and 3 (at 220.60^0) is 74.29^0 which is the largest angle difference. The two nodes are also relatively far from the depot; therefore, heuristic approach selected such point as starting angle and is found appropriate. Since $\Theta_s=180^0$ is resides between the nodes 33 and 3, the cluster formation for $\Theta_s = 220.6^0$ and $\Theta_s=180^0$ are same; and hence the CVRP outcome are also same.

Table 4.4 shows the total clusters for different starting angles (Θ_s) in variant Sweep and optimized route cost with different methods for P-n65-k10 problem. The problem has 65 nodes and total 1219 demand to be served with ten vehicle having capacity 130. From the table it is observed that total number of clusters for $\Theta_{s=}0^0$ (i.e., in standard Sweep) is 11 that is more than available vehicles. Total clusters are also 11 for $\Theta_{s=}$ 90^0. On the other hand, number of clusters equal to total vehicles (i.e., 10) for $\Theta_{s=}$ 180^0 and 270^0. It is also remarkable that final CVRP cost for 10 clusters is lower than the cases of 11. For the problem the best CVRP cost achieved

Table 4.4: Clusters for different starting angle (Θ_s) in variant Sweep and CVRP cost after route optimizing using GA, ACO, PSM and VTPSO for P-n65-k10 problem.

Θs	Clusters	CVRP Cost Before Route Optimizing	Total travel cost after optimizing with			
			GA	ACO	PSM	VTPSO
0^0	11	1142	864	933	864	864
90^0	11	1151	877	946	874	874
180^0	10	1154	**837**	900	**837**	**837**
270^0	10	1256	860	**890**	859	859
278.43^0*	10	1256	860	**890**	859	859

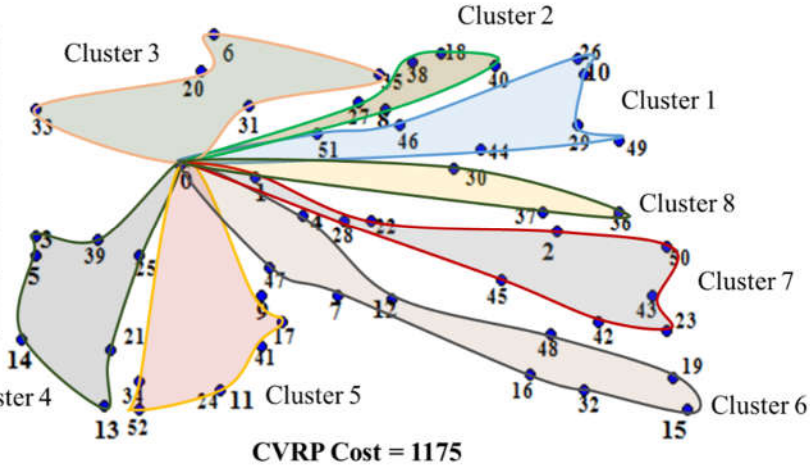
** Angle selected adaptively through proposed heuristic approach.*

(i.e., 837) for Θ_s=180^0 with GA, PSM and VTPSO. On the other hand, the heuristic approach selected starting angle is $\Theta_s = 278.43^0$ and outcome is same for fixed $\Theta_s = 270^0$ with 10 clusters. Although the outcome is inferior to best outcome with $\Theta_s = 180^0$, the outcome is better than standard Sweep with $\Theta_s = 0^0$.

Figure 4.1 is the graphical representation of the solution of A-n53-k7 problem for standard Sweep clustering (i.e., Θs=0^0). Nodes are divided into eight clusters and Cluster 8 is for remaining three nodes having total demand 29. On the other hand, Cluster 1 covers total demand of 79 although vehicle capacity 100. The CVRP costs route optimization with GA and ACO are 1175 and 1212, respectively. On the other hand, PSM and VTPSO gave same solution with CVRP cost 1174 as shown in Fig. 4.1(c). The reason for worst CVRP cost with ACO might be inclination with pheromone in ACO and solutions for Cluster 4 and Cluster 6 are bad with respect to other methods. On the other hand, slightly different solution of GA from PSM/VTPSO is shown for Cluster 6.

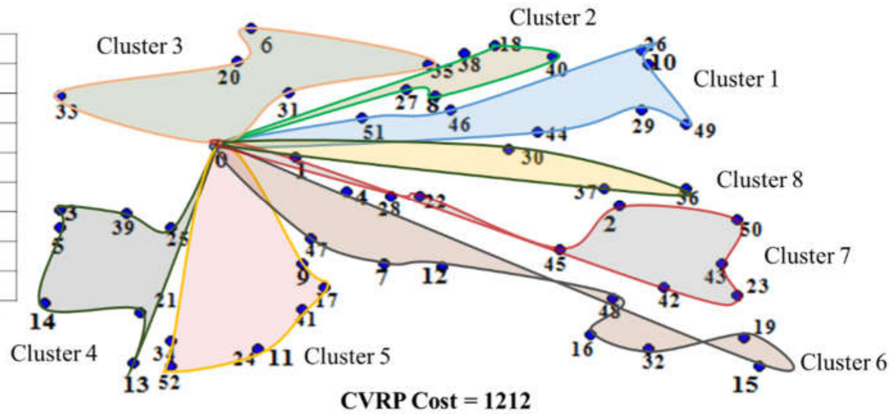
Figure 4.2 is the graphical representation of the solution of A-n53-k7 problem for variant Sweep clustering with adaptively selected $\Theta_s = 220.6^0$ or fixed Θ_s=180^0. In this case total demands are fulfilled by seven clusters that is equal to number of vehicles. Among the four route optimization methods, CVRP cost with ACO is the worst and the value is 1131. Similar to standard Sweep, it achieved worse solution for Cluster 4 and Cluster 6. The best CVRP solution for the problem is achieved by PSM and VTPSO and achieved CVRP cost is 1090. On the other hand, GA is showed competitive result with PSM/VTPSO showing different result only for Cluster 6 and CVRP cost 1091. Finally, the comparative description with graphical representation of Figs. 4.1 and 4.2 clearly identified the proficiency of proposed variant Sweep over standard Sweep.

Cluster	Demand	Cost
1	79	153
2	80	109
3	96	125
4	100	140
5	93	138
6	89	201
7	98	178
8	29	131
Total	**664**	**1175**



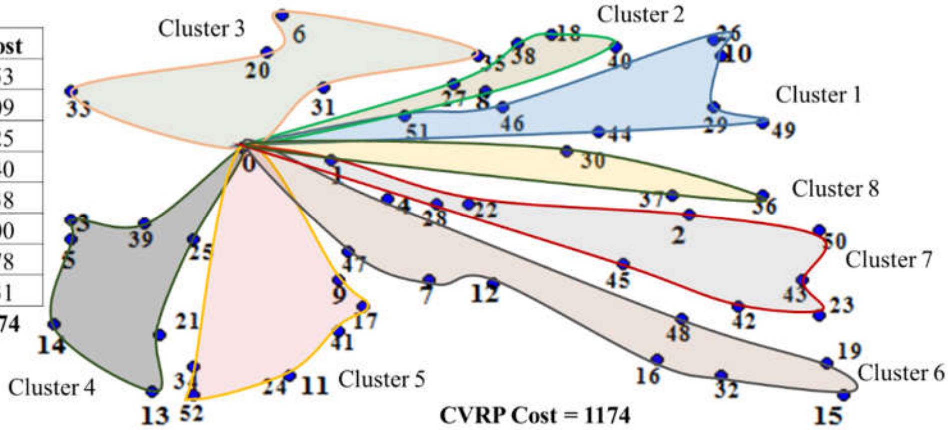
(a) Route optimization using GA.

Cluster	Demand	Cost
1	79	153
2	80	110
3	96	125
4	100	155
5	93	138
6	89	210
7	98	190
8	29	131
Total	**664**	**1212**



(b) Route optimization using ACO.

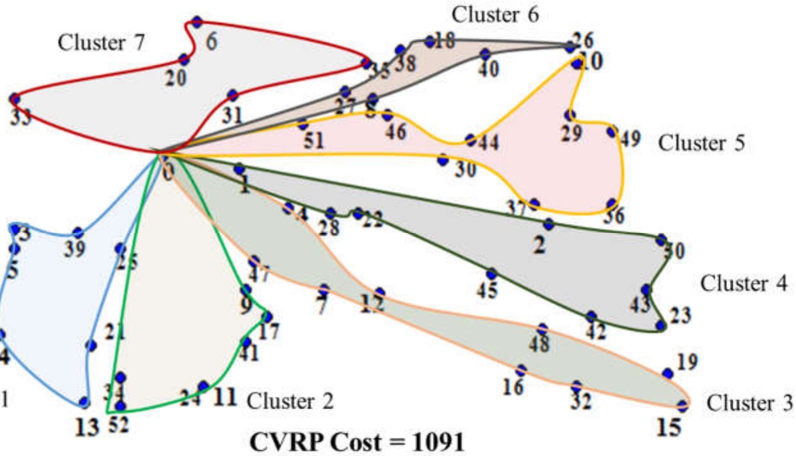
Cluster	Demand	Cost
1	79	153
2	80	109
3	96	125
4	100	140
5	93	138
6	89	200
7	98	178
8	29	131
Total	**664**	**1174**



(c) Route optimization using PSM and VTPSO.

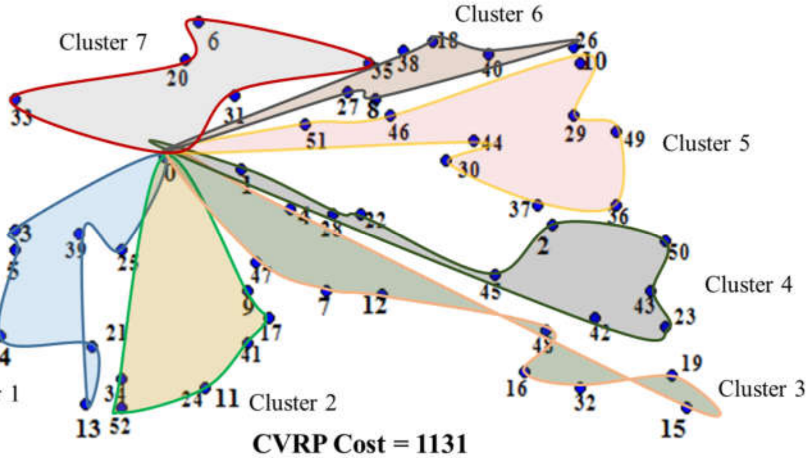
Fig. 4.1: Graphical representation of A-n53-k7 solution with standard Sweep clustering (i.e., $\Theta s=0^0$).

Cluster	Demand	Cost
1	100	140
2	93	138
3	89	201
4	98	178
5	90	177
6	98	132
7	96	125
Total	**664**	**1091**



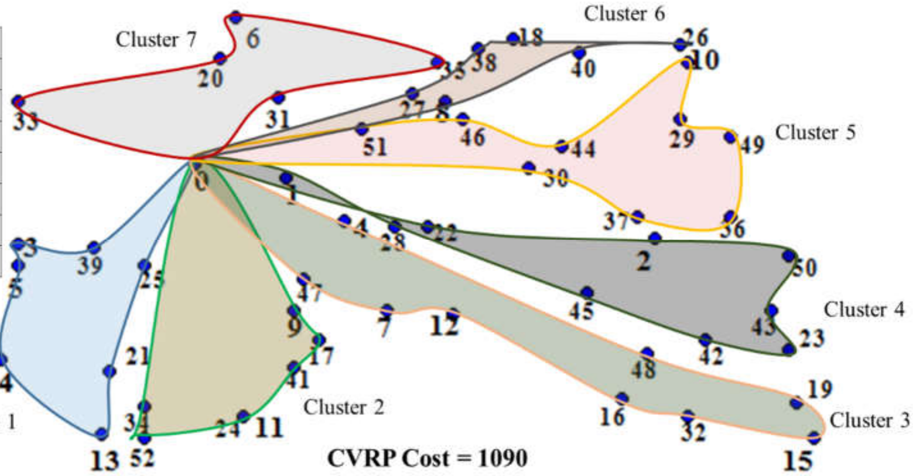
(a) Route optimization using GA.

Cluster	Demand	Cost
1	100	154
2	93	138
3	89	210
4	98	190
5	90	180
6	98	134
7	96	125
Total	**664**	**1131**



(b) Route optimization using ACO.

Cluster	Demand	Cost
1	100	140
2	93	138
3	89	200
4	98	178
5	90	177
6	98	132
7	96	125
Total	**664**	**1090**



(b) Route optimization using PSM and VTPSO.

Fig. 4.2: Graphical representation of A-n53-k7 solution with variant Sweep clustering with adaptively selected $\Theta_s = 220.6^0$ or fixed $\Theta_s=180^0$.

4.3 Experimental Results and Performance Comparison

This section first identifies the proficiency of variant Sweep clustering over standard Sweep clustering while using GA, ACO, PSO, PSM and VTPSO for route optimization. Finally the outcome of the proposed method with the prominent methods in solving benchmark CVRPs. The population size of GA, PSM and VTPSO was 100; whereas, number of ants in ACO was equal to the number of nodes assigned to a vehicle as it desired. For the fair comparison, the number of iteration was set at 200 for the algorithms. The selected parameters are not optimal values, but considered for simplicity as well as for fairness in observation.

Table 4.5 compares CVRP cost for clustering with standard Sweep and variant Sweep on A-VRP benchmark problems. Bottom of the table shows average and best/worst summary over the total 27 problems. In variant Sweep, cluster formation starting angle is problem dependent and selected through proposed heuristic approach. The starting angle is different for different problems as seen in the table. On the other hand, standard Sweep is for only Sweep clustering with $\Theta_s=0^0$.

From the Table 4.5, it is observed that most of the cases variant Sweep outperformed it corresponding standard Sweep clustering. It is notable that for a particular route optimization (e.g., GA), the outperformance of variant Sweep is only for different starting angles in variant Sweep. As an example, for A-n33-k6 problem, standard Sweep (i.e, $\Theta_s=0^0$) with GA achieved CVRP cost of 874. For the same problem the outcome of variant Sweep with adaptively selected starting angle 303.18^0 is 751. The route optimization with GA, ACO, PSM and VTPSO on variant Sweep cluster outperformed corresponding standard Sweep cluster in 20, 17, 20 and 16 out of 27 cases, respectively. Only a few cases, standard Sweep is found better than variant Sweep. As an example, for only A-n69-k9 problem with route optimization with VTPSO, standard Sweep achieved CVRP cost 1254 but variant Sweep achieved slightly larger CVRP cost and it is 1259. On the basis of average CVRP cost over 27 problems, variant Sweep is outperformed over standard Sweep for any optimization method. The average CVRP cost for standard Sweep with GA, ACO, PSM and VTPSO are 1202.04, 1224.96, 1200.93 and 1200.26, respectively. On the other hand, achieved average CVRP cost for variant Sweep with GA, ACO, PSM and VTPSO are 1169.48, 1195.33, 1169.19 and 1168.63, respectively. Among variant Sweep based methods, PSM and VTPSO outperformed GA and ACO. Finally, CVRP cost with VTPSO are found best among the methods and it showed best (i.e., minimum) CVRP costs for all 27 problems.

Table 4.5: CVRP cost comparison for clustering with standard Sweep and variant Sweep on A-VRP benchmark problems.

Sl.	Problem	CVRP Cost for Standard Sweep				Starting Angle (Θ_s)	CVRP Cost Variant Sweep with heuristically Selected Angle			
		GA	ACO	PSM	VTPSO		GA	ACO	PSM	VTPSO
1	A-n32-k5	882	897	882	882	152.02	882	897	882	882
2	A-n33-k5	788	802	788	788	195.95	698	717	698	698
3	A-n33-k6	874	877	874	874	303.18	751	758	751	751
4	A-n34-k5	867	875	867	867	203.2	785	808	785	785
5	A-n36-k5	942	965	942	942	323.13	881	917	881	881
6	A-n37-k5	795	821	795	795	248.84	756	774	756	754
7	A-n37-k6	1131	1141	1131	1131	264.29	1112	1128	1112	1112
8	A-n38-k5	857	870	857	857	148.57	813	845	813	813
9	A-n39-k5	881	912	877	877	180	877	914	877	877
10	A-n39-k6	991	997	997	991	246.8	978	975	972	972
11	A-n44-k6	1164	1229	1164	1164	253.3	1057	1116	1056	1056
12	A-n45-k6	1117	1138	1117	1115	138.01	1075	1081	1073	1073
13	A-n45-k7	1305	1333	1305	1305	180	1307	1339	1305	1305
14	A-n46-k7	985	1015	983	983	75.96	977	1010	975	975
15	A-n48-k7	1152	1164	1153	1152	3.18	1153	1165	1152	1152
16	A-n53-k7	1175	1212	1174	1174	220.6	1091	1131	1090	1090
17	A-n54-k7	1374	1374	1366	1361	4.09	1380	1374	1361	1361
18	A-n55-k9	1201	1192	1190	1190	318.96	1191	1192	1190	1190
19	A-n60-k9	1554	1602	1552	1552	170.54	1503	1528	1505	1503
20	A-n61-k9	1220	1238	1219	1219	333.43	1170	1186	1164	1164
21	A-n62-k8	1534	1560	1532	1532	263.66	1409	1435	1409	1408
22	A-n63-k9	1825	1852	1825	1823	153.43	1824	1852	1823	1823
23	A-n63-k10	1480	1511	1477	1477	6.34	1477	1511	1480	1477
24	A-n64-k9	1598	1628	1598	1598	94.57	1598	1628	1598	1598
25	A-n65-k9	1369	1394	1368	1368	237.99	1320	1327	1317	1317
26	A-n69-k9	1258	1280	1254	1254	352.09	1269	1275	1259	1259
27	A-n80-k10	2136	2195	2138	2136	149.04	2137	2195	2136	2136
Average		**1202.04**	**1224.96**	**1200.93**	**1200.26**		**1165.59**	**1188.07**	**1163.7**	**1163.41**
Outperformance of variant Sweep over corresponding Standard Sweep based method							**20**	**17**	**20**	**16**
Best Count							**12**	**0**	**23**	**27**

Table 4.6 compares CVRP cost for clustering with standard Sweep and variant Sweep on P-VRP benchmark problems. Bottom of the table shows summary of result presented for 24 problems. From the table it is observed that most of the cases variant Sweep outperformed it corresponding standard Sweep clustering. A notable difference in the P-VRP problems from A-VRP problems of Table 4.5 is that adaptively selected starting angle is same for several problems. Examination on the data revealed that such problems have similar type of node coordinates. On the basis of average CVRP cost over 24 problems, variant Sweep is

Table 4.6: CVRP cost comparison for clustering with standard Sweep and Variant Sweep on P-VRP benchmark problems.

Sl.	Problem	CVRP Cost for Standard Sweep				Starting Angle (Θ_s)	CVRP Cost Variant Sweep with heuristically Selected Angle			
		GA	**ACO**	**PSM**	**VTPSO**		**GA**	**ACO**	**PSM**	**VTPSO**
1	P-n16-k8	545	545	545	545	335.1	549	554	549	549
2	P-n19-k2	239	242	236	236	335.1	246	246	246	246
3	P-n20-k2	238	254	238	238	335.1	249	249	249	249
4	P-n21-k2	241	261	238	238	335.1	211	217	211	211
5	P-n22-k2	237	264	243	237	335.1	216	223	216	216
6	P-n22-k8	668	668	668	668	238.39	633	633	633	633
7	P-n23-k8	687	687	687	687	333.43	634	636	634	634
8	P-n40-k5	495	509	492	492	119.48	492	504	483	483
9	P-n45-k5	530	569	528	528	119.48	524	556	524	524
10	P-n50-k7	585	620	585	585	278.43	589	599	583	583
11	P-n50-k8	692	709	690	690	278.43	677	713	677	677
12	P-n50-k10	783	794	783	783	278.43	783	793	783	783
13	P-n51-k10	804	827	804	804	208.3	804	822	802	802
14	P-n55-k7	620	626	602	602	278.43	595	619	595	595
15	P-n55-k8	613	634	614	609	242.59	589	608	589	586
16	P-n55-k10	742	757	742	742	278.43	745	767	745	745
17	P-n55-k15	1133	1140	1133	1133	278.43	1099	1106	1099	1099
18	P-n60-k10	835	845	835	835	278.43	830	848	830	830
19	P-n60-k15	1092	1106	1092	1092	278.43	1119	1136	1119	1119
20	P-n65-k10	867	912	864	864	278.43	859	880	859	859
21	P-n70-k10	900	924	900	900	278.43	914	951	911	911
22	P-n76-k4	645	641	627	605	104.04	681	630	658	612
23	P-n76-k5	702	683	679	655	144.16	689	675	662	647
24	P-n101-k4	788	740	761	721	115.46	766	772	785	699
Average		**653.38**	**664.88**	**649.42**	**645.38**		**645.54**	**655.71**	**643.42**	**637.17**
Outperformance of variant Sweep over corresponding Standard Sweep based method							**14**	**16**	**15**	**16**
Best Count							**16**	**3**	**20**	**24**

outperformed over standard Sweep for any optimization method. The average CVRP cost for standard Sweep with GA, ACO, PSM and VTPSO are 653.38, 664.88, 649.42 and 645.38, respectively. On the other hand, achieved average CVRP cost for variant Sweep with GA, ACO, PSM and VTPSO are 645.54, 655.71, 643.42 and 637.17, respectively. The route optimization with GA, ACO, PSM and VTPSO on variant Sweep cluster outperformed corresponding standard Sweep cluster in 14, 16, 15 and 16 out of 24 cases, respectively. Only a few cases, standard Sweep is found better than variant Sweep. At a glance, CVRP costs with VTPSO are found best among the methods and PSM is shown competitive to VTPSO: VTPSO and PSM showed minimum CVRP cost for all 24 problems and 20 cases, respectively.

To identify the proficiency of proposed variant Sweep based approach its outcome have been compared with prominent CVRP methods. Among the selected methods, hybrid heuristic approach (HHA) [10], Sweep + Cluster Adjustment [8] and Sweep nearest [23] are also used Sweep based clustering to assign nodes to different vehicles but followed different approaches for route generation of individual vehicles. HHA [10] is the most recent CVRP method which used nearest neighbor method for route optimization. Centroid-based 3-phase [8] method is also considered in result comparison because it also found an effective method to solve similar benchmark CVRPs. The method follows three different steps: cluster formation with centroid based approach from the farthest point, centroid based cluster adjustment and finally route generation using Lin-Kernighan heuristic method.

Table 4.7 and Table 4.8 compare outcome of variant Sweep based method with the selected exiting methods in solving A-VRP and P-VRP benchmark problems. In the comparison Variant Sweep+VTPSO is considered as a proposed method since it outperformed others variant Sweep based methods and the results presented for a particular problem is same of Table 4.5 and Table 4.6. On the other hand, presented results of the existing methods are the reported results in corresponding papers. If results are not available for problems with a particular exiting method then those are marked as '-'. The best (i.e., minimum) CVRP cost among the five methods for a particular problem is marked as bold face type. Bottom of a table also shows pairwise Win/Draw/Loss summary among the methods for better understanding. According to Table 4.7, Centroid-based 3-phase is the overall best and HHA is the worst showing average CVRP cost of 1134.67 and 1310.11, respectively. On the other hand, proposed Variant Sweep+VTPSO is shown competitive to Centroid-based 3-phase showing average CVRP cost 1168.63. The proposed method showed best CVRP solution for five cases and outperformed Centroid-based 3-phase for 10 cases out of 27 cases. More interesting, the proposed method outperformed HHA, Sweep + Cluster Adjust and Sweep Nearest for 27, 15 and 10 cases, respectively.

Table 4.7: CVRP cost comparison with existing methods on A-VRP benchmark problems.

Sl.	Problem	HHA [10]	Centroid-based 3-phase [8]	Sweep + Cluster Adjust [8]	Sweep Nearest [23]	Variant Sweep + VTPSO
1	A-n32-k5	1012	881	872	**853**	882
2	A-n33-k5	847	728	788	702	**698**
3	A-n33-k6	919	770	829	767	**751**
4	A-n34-k5	933	812	852	803	**785**
5	A-n36-k5	1126	**814**	884	840	881
6	A-n37-k5	876	756	**734**	797	754
7	A-n37-k6	1180	1027	1050	**966**	1112
8	A-n38-k5	920	819	874	**801**	813
9	A-n39-k5	1147	**864**	971	886	877
10	A-n39-k6	1065	**881**	966	-	972
11	A-n44-k6	1356	1037	1092	**1020**	1056
12	A-n45-k6	1210	1040	1043	**991**	1073
13	A-n45-k7	1361	1288	1281	**1235**	1305
14	A-n46-k7	1071	992	1013	1022	**975**
15	A-n48-k7	1292	1145	**1143**	1181	1152
16	A-n53-k7	1261	1117	1116	-	**1090**
17	A-n54-k7	1414	**1209**	1320	-	1361
18	A-n55-k9	1317	1155	1192	**1134**	1190
19	A-n60-k9	1733	**1430**	1574	1446	1503
20	A-n61-k9	1285	1201	1184	**1158**	1164
21	A-n62-k8	1604	1470	1559	**1392**	1408
22	A-n63-k9	2001	1766	1823	**1763**	1823
23	A-n63-k10	1542	**1405**	1523	1475	1477
24	A-n64-k9	1821	1587	1597	**1586**	1598
25	A-n65-k9	1429	**1276**	1351	1299	1317
26	A-n69-k9	1333	1283	1254	**1225**	1259
27	A-n80-k10	2318	**1883**	2014	1896	2136
	Average	**1310.11**	**1134.67**	**1181.44**	**1134.92**	**1163.41**
	Best/Worst	**0/27**	**8/0**	**2/0**	**12/0**	**5/0**
			Pairwise Win/Draw/Loos Summary			
	HHA	-	27/0/0	27/0/0	24/0/0	27/0/0
	Centroid-based 3-phase		-	7/0/20	15/0/9	10/0/17
	Sweep + Cluster Adjust			-	21/0/3	15/1/11
	Sweep Nearest				-	10/0/17

The comparative results in Table 4.8 identified the proposed Variant Sweep + VTPSO is the best for P-VRP benchmark problems. The proposed method is shown the best for 10 cases out of 24 cases and achieved average cost of 637.17. The proposed method outperformed HHA, Centroid-based 3-phase, Sweep + Cluster Adjust, Sweep Nearest on 23, 12, 13 and 6 cases, respectively, out of 24 cases. It is notables that Sweep Nearest tested only 10 problems. Between two exiting Sweep based methods, HHA only outperformed proposed method only

Table 4.8: CVRP cost comparison with existing methods on P-VRP benchmark problems.

Sl.	Problem	HHA [10]	Centroid-based 3-phase [8]	Sweep + Cluster Adjust [8]	Sweep Nearest [23]	Variant Sweep + VTPSO
1	P-n16-k8	546	497	568	**463**	549
2	P-n19-k2	253	256	**236**	264	246
3	P-n20-k2	267	240	238	**217**	249
4	P-n21-k2	288	240	238	**211**	**211**
5	P-n22-k2	274	245	237	219	**216**
6	P-n22-k8	667	672	687	721	**633**
7	P-n23-k8	743	703	645	**558**	634
8	P-n40-k5	563	505	499	516	**483**
9	P-n45-k5	662	533	525	-	**524**
10	P-n50-k7	647	**583**	585	-	**583**
11	P-n50-k8	721	**669**	675	-	677
12	P-n50-k10	808	**740**	779	-	783
13	P-n51-k10	857	**779**	806	-	802
14	P-n55-k7	679	610	611	-	**595**
15	P-n55-k8	690	654	601	-	**586**
16	P-n55-k10	832	749	763	-	**745**
17	P-n55-k15	1180	**1022**	1056	-	1099
18	P-n60-k10	896	**786**	823	-	830
19	P-n60-k15	1159	**1006**	1086	-	1119
20	P-n65-k10	964	**836**	856	-	859
21	P-n70-k10	989	**891**	902	-	911
22	P-n76-k4	753	685	**603**	690	612
23	P-n76-k5	671	737	**647**	-	**647**
24	P-n101-k4	891	**698**	702	789	699
	Average	**708.33**	**639.00**	**640.33**	**464.8**	**637.17**
	Best/Worst	**0/20**	**10/1**	**3/1**	**4/2**	**10/0**
		Pairwise Win/Draw/Loos Summary				
	HHA	-	21/0/3	22/0/2	8/0/2	23/0/1
	Centroid-based 3-phase		-	10/0/14	5/0/5	12/1/11
	Sweep + Cluster Adjust		-	-	5/0/5	13/1/10
	Sweep Nearest				-	6/1/3

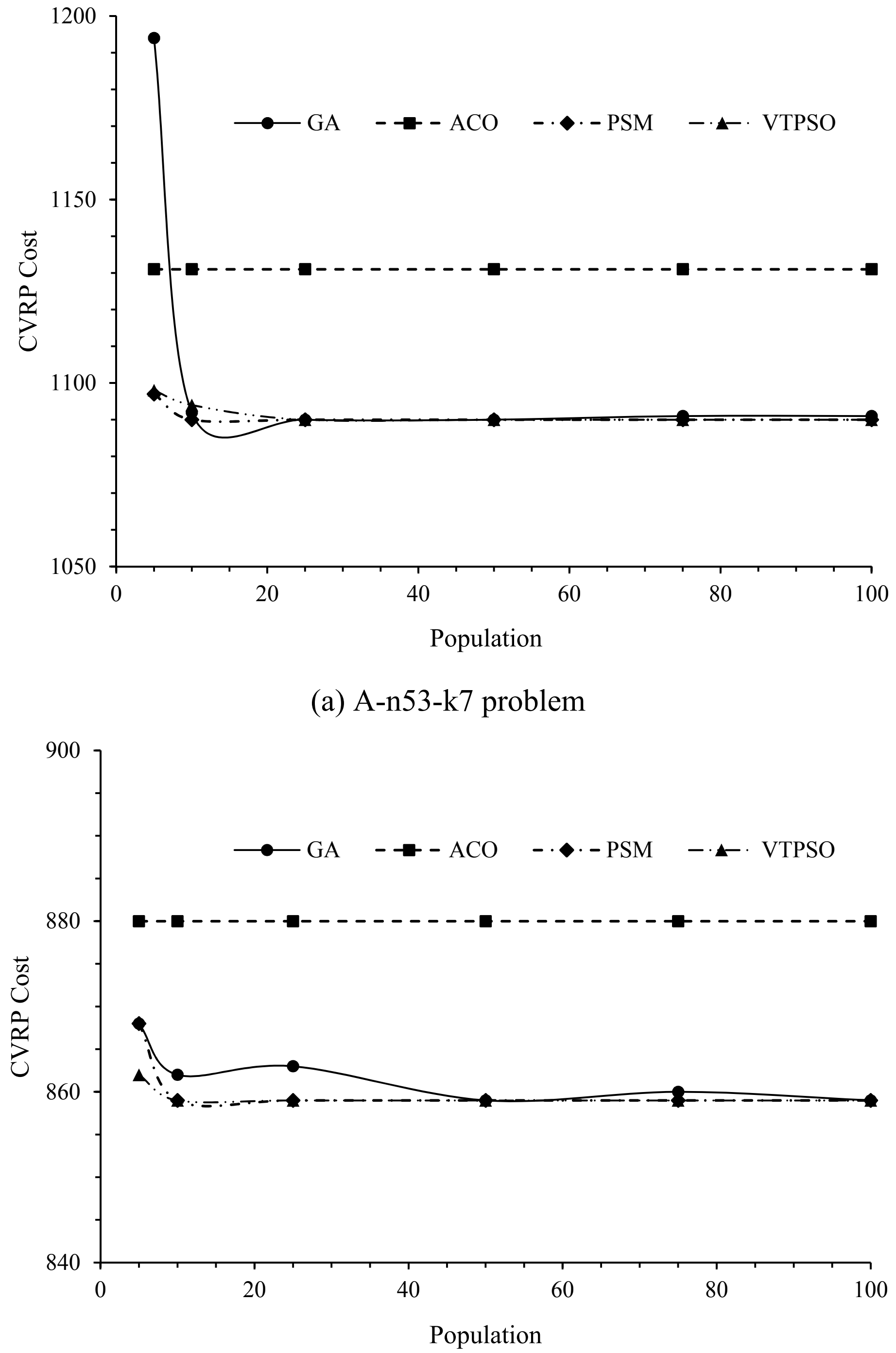
for P-n16-k8 that is very small sized problem. Finally, the proposed Variant Sweep+VTPSO is identified the proficiency of variant Sweep in clustering and VTPSO in route optimizing.

4.4 Experimental Analysis

The results presented in Table 4.5 and Table 4.6 are for fixed population and iteration in the optimization technique; and therefore it is required to investigate variation effect of population and iteration in the methods on CVRP cost. Finally, the effect of population size on route optimizing has been investigated for A-n53-k7 and P-n65-k10 problems. Population size was varied from 5 to 100 for GA, PSM and VTPSO; whereas, the number of ants in ACO was equal to the number of cities as it desired.

Fig. 4.3 shows CVRP cost (i.e., total travel cost) for population variation; experiment conducted for fixed 200 iteration for fair comparison. The number of clusters (i.e., vehicles) were 7 and 10 for A-n53-k7 and P-n65-k10 problems, respectively. From the figure it is observed that CVRP cost is invariant for ACO because population variation was not employed for it. On the other hand, GA is most sensitive with population size: CVRP cost through GA was very bad with respect to others at small population size (e.g., 5) and was competitive at larger population size. From the figure it is also observed that recent SI methods PSM and VTPSO are better than ACO and GA in population variation. At a glance, VTPSO is shown to outperform any other method for any population size and PSM is competitive to VTPSO.

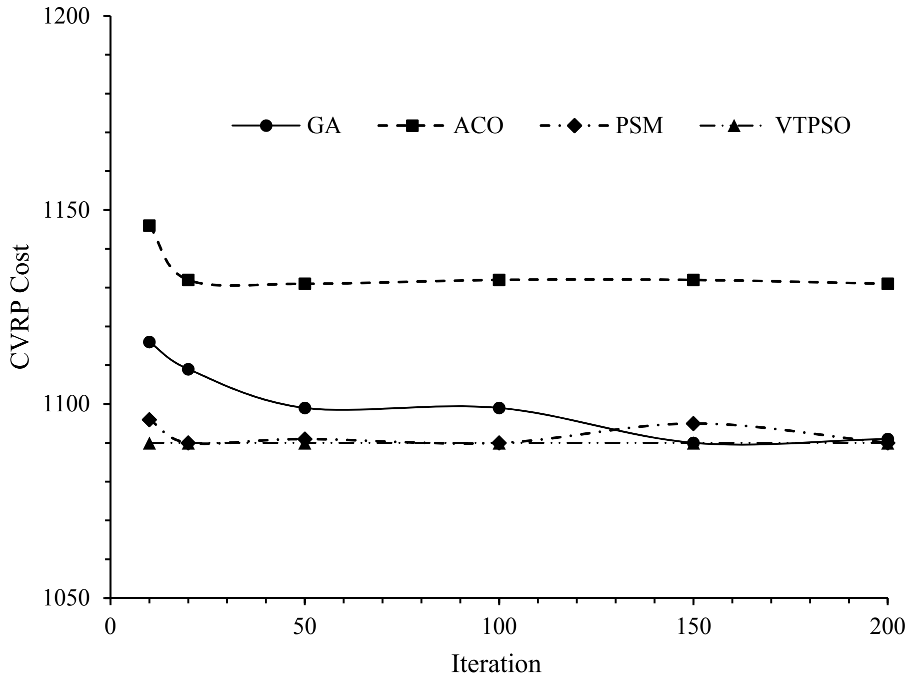
Fig. 4.4 shows CVRP cost varying iteration from 10 to 200 while population size was fixed at 100 for each of GA, PSM and VTPSO. Similar to previous experiments, the number of ants in ACO was equal to the number of nodes in a cluster. From the figure it is observed that CVRP cost is higher at small iteration (e.g., 10) and improved with iteration up to a certain level such as 100, in general. However GA is shown very worse than others for small number of iteration. Form the figure it also observed that recent SI methods PSM and VTPSO are better than ACO and GA in iteration variation.



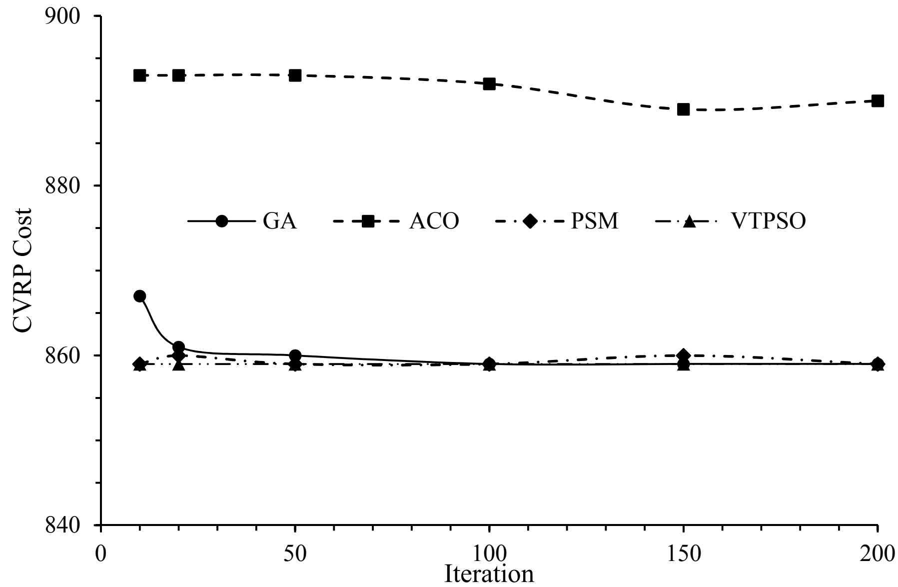
(a) A-n53-k7 problem

(b) P-n65-k10 problem

Fig. 4.3: Effect of population size in route optimization method on CVRP cost.



(a) A-n53-k7 problem



(b) P-n65-k10 problem

Fig. 4.4: Effect of iteration in route optimization method on CVRP cost.

Chapter V
Conclusions

Optimization has been an active area of research for several decades. Capacitated Vehicle Routing Problem (CVRP) is the most popular combinatorial optimization problem and interest grows in recent years to solve it new ways. This thesis investigated a CVRP solving method with a variant of Sweep to assign nodes into different clusters and prominent TSP optimization method to generate individual vehicle's route for individual cluster's nodes. This chapter will now give a short summary of the main points described in this thesis. Also, it discusses possible future works based on the outcome of the present work.

5.1 Achievements

A popular way of solving CVRP is to cluster the nodes according to vehicles using Sweep algorithm first and then generate route for each vehicle with TSP algorithm. In general, Sweep cluster construction starts from the node having lowest polar angle. In this study, a variant Sweep clustering is investigated which starts cluster formation from different starting angle to achieve better CVRP solution. A heuristic based adaptive method is developed to select appropriate cluster formation starting angle. Finally, GA, ACO, PSM and VTPSO are applied to generate optimal route with the clusters. The experimental result on the benchmark problems revealed that different starting angles have positive effect on Sweep clustering. On the other hand, recent SI based methods PSM and VTPSO are found better than GA and ACO to generate vehicle routes.

5.2 Perspectives

There are several future potential directions that follow from this study. In this study, angle difference and distance from the depot are considered to select starting angle. Scheme with node demand might be interesting. Moreover, it might be interesting to incorporate SI based methods on other cluster first route second CVRP methods.

References

[1] G. B. Dantzig and J. H. Ramser, 1959, "The Track Dispatching Problem", *Management Science*, Vol. 6, No. 1, pp. 80-91, Oct., 1959.

[2] G. Laporte, M. Gendreau, J-Y. Potvin, and F. Semet., "Classical and modern heuristics for the vehicle routing problem", *International Transactions in Operational Research*, Vol. 7, 285-300, 2000.

[3] P. Toth, D. Vigo, "*The Vehicle Routing Problem*", SIAM Monographs on Discr. Math. Appl. 9, 2001

[4] T. Caric, A. Galić, J. Fosin, *et al.* "A Modelling and Optimization Framework for Real-World Vehicle Routing Problems", *Vehicle Routing Problem*, ISBN 978-953-7619-09-1, Austria: In-Teh, 2008.

[5] S. N. Kumar, R. Panneerselvam, "A Survey on the Vehicle Routing Problem and Its Variants", *Intelligent Information Management*, Vol. 4, pp. 66-74, 2012.

[6] Report on overview of the Vehicle Routing Problem. Available: https://info.maths.ed.ac.uk/assets/files/Projects/17.pdf

[7] F. Takes, "Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem", Msc. Thesis, Leiden University, The Netherlands, February 26, 2010.

[8] K. Shin, "A Centroid-Based Heuristic Algorithm for the Capacitated Vehicle Routing Problem", *Computing and Informatics*, vol. 30, pp. 721–732, 2011.

[9] R. Z. Farahani, "The Vehicle-Routing Problem," Logistics Operations and Management: Concepts and Models, ISBN: 978-0-12-385202-1, USA: Elsevier Insights, 2011.

[10] M. M. A. Aziz, H. A. El-Ghareeb and M. S. M. Ksasy, "Hybrid Heuristic Algorithm for solving Capacitated Vehicle Routing Problem", *International Journal of Computers & Technology*, Vol. 12, no.9, pp. 3845-3851, 2014.

[11] G. Vaira, "Genetic Algorithm for Vehicle Routing Problem," PhD. Thesis, Vilnius University, Italy, 2014.

[12] M. Islam, S. Ghosh, M. Rahman, "Solving Capacitated Vehicle Routing Problem by Using Heuristic Approaches: A Case Study", *Journal of Modern Science and Technology,* Vol. 3, pp.135-146, March 2015.

[13] N. Christofides, A. Mingozzi, P. Toth, "Exact algorithms for the vehicle routing problem based on the spanning tree and shortest path relaxations", *Math. Programming*, Vol. 20, pp. 255–282, 1981.

[14] J. Cordeau, G. Laporte, M. W.P. Savelsbergh, D. Vigo, "Short-Haul Routing". Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.122.9096

[15] M. Radiy, "A Hybrid Method for Capacitated Vehicle Routing Problem", PhD. Thesis, Curtin University of Technology, Australia, March 2010.

[16] Examples and Benchmark for CVRP: http://neo.lcc.uma.es/radi-aeb/WebVRP.

[17] B. E. Gillet and L. R. Miller. "A Heuristic Algorithm for the Vehicle Dispatch Problem". *Operations Research*, Vol. 22, pp. 340-349, 1974.

[18] G. W. Nurcahyo, R. A. Alias, S. M. Shamsuddin and M. N. Md. Sap, "Sweep Algorithm in vehicle routing problem for public Transport", *Journal Antarabangsa (Teknologi Maklumat)*, vol. 2, pp. 51-64, 2002.

[19] N. Suthikarnnarunai, "A Sweep Algorithm for the Mix Fleet Vehicle Routing Problem", Proceedings of the International Multi-Conference of Engineers and Computer Scientists 2008 Vol II, IMECS 2008, 19-21 March, 2008, Hong Kong.

[20] A.Boonkleaw, N. Suthikarnnarunai, PhD, R. Srinon, PhD, "Strategic Planning and Vehicle Routing Algorithm for Newspaper Delivery Problem: Case study of Morning Newspaper, Bangkok, Thailand", Proceedings of the World Congress on Engineering and Computer Science, Vol II, 2009.

[21] M. Yousefikhoshbakht and E. Khorram, "Solving the vehicle routing problem by a hybrid meta-heuristic algorithm", *Journal of Industrial Engineering International*, vol. 8(11), 2011.

[22] S. R.Venkatesan, D. Logendran, D. Chandramohan, "Optimization of Capacitated Vehicle Routing Problem using PSO", *International Journal of Engineering Science and Technology (IJEST)*, Vol 3, pp. 7469-7477, October 2011.

[23] Byungsoo Na, Yeowoon Jun & Byung-In Kim, "Some extensions to the sweep algorithm" *Int J Adv Manuf Technol*, Vol. 56, pp. 1057–1067, 2011.

[24] D. E. Goldberg, 'Genetic Algorithm in Search, Optimization and Machine Learning', New York: Addison – Wesley, 1989.

[25] H. M. Harmanani, D. Azar and N. G. Helal, "A Simulated Annealing Algorithm for the Capacitated Vehicle Routing Problem", in *Proceedings of the ISCA 26th International Conference on Computers and Their Applications*, March 2011.

[26] T. Stutzle and M. Dorigo, "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances", *International Series in Operations Research and Management Science*, Kluwer, 2001.

[27] Y. F. Liao, D. H. Yau and C. L. Chen, "Evolutionary algorithm to traveling salesman problems," *Computers & Mathematics with Applications,* vol. 64, no. 5, pp. 788–797, 2012.

[28] A. Subramanian, “Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems”, PhD. Thesis, Universidade Federal Fluminense, Niteroi, March 05, 2012.

[29] M. M. Tavakoli, Ashkan Sami, “Particle Swarm Optimization in Solving Capacitated Vehicle Routing Problem”, *Bulletin of Electrical Engineering and Informatics*, Vol. 2, No. 4, pp. 252-257, ISSN: 2089-3191, December 2013.

[30] M. Wen, “Rich Vehicle Routing Problems and Applications”, PhD. Thesis, Technical University of Denmark, 2010.

[31] Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D., Rinaldi, G., “Computational results with a branch and cut code for the capacitated vehicle routing problem”, 1998.Available:https://www.researchgate.net/publication/239062886_Computational_results_with_a_branch_and_cut_code_for_the_capacitated_vehicle_routing_problem.

[32] Xiaoyan Li, “Capacitated Vehicle Routing Problem with Time Windows: A Case Study on Pickup of Dietary Products in Nonprofit Organization”, Msc. Thesis, Arizona State University, December 2015.

[33] M. F. Demiral, I. Gungor, K. O. Oruc, “Optimization at Service Vehicle Routing and a Case Study of Isparta, Turkey ”, International Conference on Management and Economics: Current Issues in Emerging Economies in Global Perspective, pp 374-387, ICME 2008. Available: http://www.epoka.edu.al/new/icme/27.pdf

[34] S. Amini, H. Javanshir & R. T. Moghaddam, “A PSO Approach for Solving VRPTW With Real Case Study”, *IJRRAS*, August 2010.

[35] J. Faulin, A. Juan, F. Lera, S. Grasman, “Solving the Capacitated Vehicle Routing Problem with Environmental Criteria Based on Real Estimations in Road Transportation: A Case Study”, *Procedia Social and Behavioral Sciences*, Vol. 20, pp 323–334, 2011.

[36] Classical CVRP solving techniques. http://wiki4city.ieis.tue.nl/index.php?title=VRP.

[37] G. Laporte, Y. Nobert and S. Taillefer, “A Branch-and-Bound Algorithm for the Asymmetrical Distance-Constrained Vehicle Routing Problem”, *Math Modeling*, Vol. 9. No. 12. pp. 857-868, 1987.

[38] D. Vigo, VRPLIB: a vehicle routing problem library, Technical Report OR/99/9, D.E.I.S., University of Bologna, Italy, 1999.

[39] G.H. Dastghaibifard, E. Ansari, S.M. Sheykhalishahi, A. Bavandpouri, E. Ashoor, “A Parallel Branch and Bound Algorithm for Vehicle Routing Problem ”, Proceedings of the International MultiConference of Engineers and Computer Scientists Vol. 2, 19-21 March, Hong Kong 2008.

[40] Simplex method. http://mathworld.wolfram.com/SimplexMethod.html.

[41] Linear programming relaxation. Available: https://en.wikipedia.org /wiki/Linear_progamming_relaxation

[42] LP Relaxation and Rounding. http://pages.cs.wisc.edu/~shuchi/courses/787-F09/scribe notes/lec10.pdf.

[43] Branch and Cut method. https://en.wikipedia.org/wiki/Branch_and_cut.

[44] J. Lysgaard, A.N. Letchford, and R.W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Math. Program, Vol.* 2, No. 1000, pp. 23–445, 2004.

[45] Jens Lysgaard, "Clarke & Wright's Savings Algorithm", Department of Management Science and Logistics, The Aarhus School of Business, 1997.

[46] A. Poot & G. Kant, "A Savings Based Method for Real-Life Vehicle Routing Problems", Econometric Institute Report EI 9938/A, August 1, 1999.

[47] Clarke, G. & Wright, J.W., 1964, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research*, Vol. 12, pp. 568-581.

[48] M. Straka, S. Grescovicova, R. Lenort, P. Besta, "Methodology for Optimization of Transport Plans in Tesco Distribution Center Presov", Carpathian Logistics Congress, 7th - 9th November, Jeseník, Czech Republic, EU, 2012.

[49] T. Doyuran and B. Çatay, "A robust enhancement to the Clarke-Wright savings algorithm", 2011. Technical notes, Available: http://research.sabanciuniv.edu/ 16244/1/Robust_Enhancement_to_CW_Algorithm.pdf

[50] T. Pichpibula, R. Kawtummachai, "An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem", *Science Asia*, Vol. 38, pp. 307–318, 2012.

[51] K. Altinkemer and B. Gavish., "Parallel savings based heuristic for the delivery problem", *Operations Research,* 39:456-469, 1991.

[52] M. Vidovic, "Matching Algorithm for the Vehicle Routing in Containers Delivery Collecting Problems", 7th Balkan Conference on Operational Research, Romania , May 2005.

[53] M. L. Fisher and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing". *Networks*, Vol. 11, pp. 109-124, 1981.

[54] C. Rainwater, J. Geunes, H. E. Romeijn, "The generalized assignment problem with flexible jobs", *Discrete Applied Mathematics*, Vol. 157, pp. 49–67, 2009.

[55] S. Han and Y. Tabata, "A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Controlling Lethal Gene," *Asia Pacific Management Review*, vol. 7, no. 3, pp. 405-426, 2002.

[56] S.R. Venkatesan , Z. Basha , G. Ofgaa, "The New Joined Sweep and Clark and Wright Algorithm for Clustering of Capacitated Vehicle Routing Problem", *International*

Journal of Advances in Electrical and Electronics Engineering", Vol 3, No. 3, pp. 169-174, 2013.

[57] J. Renaud, F. F. Boctor, G. Laporte, "An Improved Petal Heuristics for the Vehicle Routing Problem", *Journal of the operational research society*, Vol. 47, 329-336, 1996.

[58] D. M. Ryan, C. Hjorring and F. Glover, "Extensions of the Petal Method for Vehicle Routing", *The Journal of the Operational Research Society*, Vol. 44, No. 3, pp. 289-296, March 1993.

[59] P.C. Chu AND J.E. Beasley, "Constraint Handling in Genetic Algorithms: The Set Partitioning Problem", *Journal of Heuristics*, Vol.11, pp. 323–357, 1998.

[60] C. Hjorring, "Using Genetic Algorithms and the Petal Method to Solve Vehicle Routing Problems ", *proceeding of NZ Operational Research conference*, 1993.

[61] J. E. Beasley, "Route-First Cluster Second Method for Vehicle Routing", *International Journal of Management Science*,Vol.11,No. 4, pp.403-408, 1983.

[62] T. K. Ralphs, "Parallel branch and cut for capacitated vehicle routing", *Parallel Computing*, Vol.29, pp.607–629, 2003.

[63] M. A. H. Akhand, P. C. Shill, Md. Forhad Hossain, A. B. M. Junaed, and K. Murase, "Producer-Scrounger Method to Solve Traveling Salesman Problem", *I. J. Intelligent Systems and Applications*, vol. 7, no. 3, pp. 29-36, 2015.

[64] M. A. H. Akhand, S. Akter, M. A. Rashid and S. B. Yaakob, "Velocity Tentative PSO: An Optimal Velocity Implementation based Particle Swarm Optimization to Solve Traveling Salesman Problem", IAENG International Journal of Computer Science, vol. 42, no.3, pp 221-232, 2015.

[65] P. Toth, D. Vigo, "Models, relaxations and exact approaches for the capacitated vehicle routing problem", *Discrete Applied Mathematics*, pp. 487 – 512, 2002.

[66] R. Baldacci., E. Hadjiconstantinou, A. Mingozzi, "An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation", *Operations Research*, Vol. 52, pp.723–738, 2004.

Publications Resulting from the Thesis

1. M. A. H. Akhand, Zahrul Jannat, Tanzima Sultana and Al-Mahmud, "Optimization of Capacitated Vehicle Routing Problem using Producer-Scrounger Method," *in Proc. of 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE 2015)*, BUET, Dhaka, Bangladesh, pp. 297-300, December 19-20, 2015.

2. M. A. H. Akhand, Zahrul Jannat, Tanzima Sultana and Al-Mahmud, "Solving Capacitated Vehicle Routing Problem with Route Optimization using Swarm Intelligence," *in Proc. of 2015 2nd International Conference on Electrical Information and Communication Technology (EICT)*, Khulna, Bangladesh, pp. 117-122, December 10-12,2015.