ISSN: 0974-6471, Vol. 10, No. (3) 2017, Pg. 549-562

Oriental Journal of Computer Science and Technology

Journal Website: www.computerscijournal.org

Constructive and Clustering Methods to Solve Capacitated Vehicle Routing Problem

M. A. H. AKHAND, TANZIMA SULTANA, M. I. R. SHUVO and AL-MAHMUD

Dept. of Computer Science and EngineeringKhulna University of Engineering & Technology (KUET)Khulna-9203, Bangladesh.

Abstract

Vehicle Routing Problem (VRP) is a real life constraint satisfaction problem to find minimal travel distances of vehicles to serve customers. Capacitated VRP (CVRP) is the simplest form of VRP considering vehicle capacity constraint. Constructive and clustering are the two popular approaches to solve CVRP. A constructive approach creates routes and attempts to minimize the cost at the same time. Clarke and Wright's Savings algorithm is a popular constructive method based on savings heuristic. On the other hand, a clustering based method first assigns nodes into vehicle wise cluster and then generates route for each vehicle. Sweep algorithm and its variants and Fisher and Jaikumar algorithm are popular among clustering methods. Route generation is a traveling salesman problem (TSP) and any TSP optimization method is useful for this purpose. In this study, popular constructive and clustering methods are studied, implemented and compared outcomes in solving a suite of benchmark CVRPs. For route optimization, Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Velocity Tentative Particle Swarm Optimization (VTPSO) are employed in this study which are popular nature inspired optimization techniques for solving TSP. Experimental results revealed that parallel Savings is better than series Savings in constructive method. On the other hand, Sweep Reference Point using every stop (SRE) is the best among clustering based techniques.

Article History

Received: 21 July 2017
Accepted:17 August 2017

Keywords

Capacitated Vehicle Routing Problem, Constructive method, Clustering method, Clarke and Wright's Savings Algorithm, Sweep Algorithm, Fisher and Jaikumar Algorithm, Genetic Algorithm, Ant Colony Optimization and Velocity Tentative Particle Swarm Optimization.

Introduction

Vehicle Routing Problem (VRP) is a real life constraint satisfaction problem to find minimal travel distances of vehicles to serve customers[1]. Capacitated VRP (CVRP) is the simplest form of VRP considering equal vehicle capacity constraint[2]. In CVRP, all customers have known demands and known locations for the delivery. The delivery for a customer cannot be split. In other words, the demand of a customer must be satisfied via only one visit. Goods are delivered from single depot and the vehicles return to the depot after serving

CONTACT Dr. Muhammad Aminul Haque Akhand ✉ akhandkuet@yahoo.com Dept. of Computer Science and EngineeringKhulna University of Engineering & Technology (KUET)Khulna-9203, Bangladesh.

To link to this article: http://dx.doi.org/10.13005/ojcst/10.03.02

their assigned customers. The delivery or unloading time may or may not be considered. The objective of CVRP is to minimize the total travelling distance for all vehicles.

Various ways have been investigated for solving the CVRP. Nearly all of them are heuristics because no exact algorithm can be guaranteed to find optimal tours within reasonable computing time when the problem is large. A heuristic approach does not explore the entire search space rather tries to find an optimal solution based on the available information of the problem. CVRP solving heuristic approaches are categorized as constructive and clustering methods. Constructive methods build a feasible solution gradually while keeping an eye on solution cost, but it may not contain an improvement or optimization phase. Some well-known algorithms of constructive methods are - Clarke and Wright's Savings Algorithm [3, 4, 5, 6, 7], Matching Based Algorithm and Multi-route Improvement Heuristics[8]. On the other hand, clustering methods solve problem in two steps and that is why this approach is also called 2-phase or cluster first, route second method. First phase of this approach uses clustering algorithm to generate clusters of customers and second phase uses optimization technique to find optimum routes for each cluster generated in the first step. Sweep[9, 10] and Fisher and Jaikumar[11] are popular clustering algorithms.

This study investigates and compares performance of prominent constructive and clustering methods to solve CVRP and hence find out the best suited one. The initial clusters found from clustering methods are subjected to route optimization. For route optimization, Genetic Algorithm (GA)[12], Ant Colony Optimization (ACO)[13] and Velocity Tentative Particle Swarm Optimization (VTPSO)[14] are employed in this study which are popular nature inspired optimization techniques for solving TSP. Though constructive methods generate optimal routes, route optimization techniques are also applied to Savings algorithm for performance analysis.

The outline of the paper is as follows; Section II explains prominent constructive and clustering methods briefly. Section III gives brief description of the optimization techniques. Section IV is for experimental studies in which outcomes of the selected methods are compared on a suite of benchmark CVRPs. At last, Section V gives a brief conclusion of the paper.

Constructive And Clustering Methods To Solve Cvrp

Constructive and clustering methods are popular in solving CVRPs. In this section prominent methods are explained briefly.

Constructive Method

A method in this category constructs routes as well as tries to minimize the travelled distance of the vehicles at the same time. Clarke and Wright's Savings algorithm is the most popular constructive method and which is explained below.

Clarke and Wright's Savings Algorithm

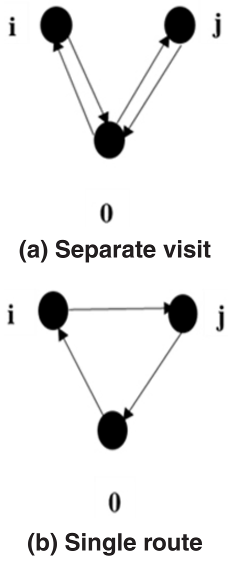
The algorithm is based on savings distance [3] which is obtained by joining two routes into one route as illustrated in Fig. 1 where point 0 represents the depot. Denoting the transportation cost between two given points i and j, total transportation costs for separate (Fig. 1(a)) and single visit (Fig. 1(b)) are shown in Eq. (1) and Eq. (2), respectively.

$$D_a = C_{0i} + C_{i0} + C_{0j} + C_{j0} \quad ...(1)$$

$$D_b = C_{0i} + C_{ij} + C_{j0} \quad ...(2)$$

Now, the distance saved (savings distance) by visiting i and j in the same route instead of separately is,

$$S_{ij} = D_a - D_b = C_{i0} + C_{0j} - C_{ij} \quad ...(3)$$



(a) Separate visit

(b) Single route

Fig. 1: Savings demonstration

Algorithm 1: Savings (Series)
Initialization

- Set k = 1.
- For all pair of nodes i,j (i≠j) calculate savings distance using Eq. (3).
- Sort the node pairs according to the decreasing order of savings distance.

Route Generation

- Create a new route Rk with the top pair of nodes (0-i-j-0).
- Process each node pair (i,j) for the current route Rk {0-n1-n2-…….nn-0},
- If one of the nodes (i,j) found in Rk, the other one is inserted if demand condition is satisfied. Remove the pair from savings list and node from node list. Then start again from the top of the list.
- Repeat Step 2 until last pair of node is processed.
- After all the node pairs have been processed,
- If both the node pair list and node list is empty then Stop.
- Else if node pair list is empty but not the node list, then go to Step 1(b).

The more the savings distance, the closer the customers are and more distant they are from depot. The customer pairs are sorted in descending order of their savings and route construction starts from top of the list. When a pair of nodes i-j is considered, if i or j is already in the route then other one is considered for insertion. There are two approaches of Savings algorithm; series/sequential and parallel.

In series approach, if a customer pair does not match, it is skipped. Each time a customer is inserted into a route, one must start a new from the top of the list as the combinations that were not viable so far, now may have become viable. It creates one route at a time and requires several pass through the customer pair list. Algorithm 1 shows the steps of the series Savings.

The parallel version of Savings creates multiple partial routes at a time. When a pair of customer is processed, it is checked against all the routes already generated instead of only the current one. If a pair doesn't matches, a new route is created with the pair, while it is skipped in series approach. After each insertion, the partial routes are considered for merging. In parallel Savings, only one pass requires through the savings pair list for route construction. The steps of the parallel Savings are shown in Algorithm 2.

Algorithm 2: Savings (Parallel)
Initialization

- Same as series Initialization (a-c).
- Create a route with the first fair of nodes (0-i-j-0).

Route Generation

Process each pair of nodes for all the routes Rp{0-n1-n2-….nn-0},

- If one of the nodes (i,j) matches with Rp, the other one is inserted if demand condition is satisfied. Remove the inserted node from node list.
- If the nodes don't match, then create a new route with pair. Set p = p+1.
- If k<= 1 then repeat Step 2.

Merge Routes

- First or last node of two routes matches (0-i-j-0 and 0-k-j-0) and the total demand of the combined route satisfy vehicle capacity, then merge the routes (0-i-j-k-0), delete one of the routes. Update k.
- Repeat Steps 2-3 until all the node pairs have been processed.
- If all the pairs are processed but the node list is not empty, go to Step 1(b). Else stop.

Clustering Method

In clustering method, the solution of the problem is divided into two phases. All the nodes are grouped into several clusters using clustering algorithms and then the routes are optimized using any optimization technique. That's why this approach is also known as 2-phase or cluster first, route second algorithm.

Sweep Algorithm

Sweep algorithm is the simplest clustering method for solving CVRP[9, 10]. Cluster formation starts from 00 and consequently advance towards 360^0 to assign all the nodes under different vehicles while maintaining vehicle capacity. This type of sweeping is called forward sweep. And in backward sweep, clustering direction is clockwise which means though clustering starts from 0^0, then it advances algorithm from

360^0 to 0^0. The general formula for calculating polar angle of the customers with respect to depot is,

$$\theta = \tan^{-1}(y/x) \qquad ...(4)$$

where
θ = Angle of a node (depot/customer).
x,y= X, Y co-ordinates of customer.

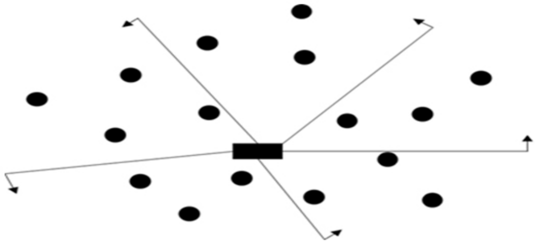


Fig. 2: Sweep clustering

Algorithm 3: Sweep
Initialization

- Compute the polar angle of each customer using Eq. (4).
- Sort the customer according to their increasing order of polar angles.

Clustering

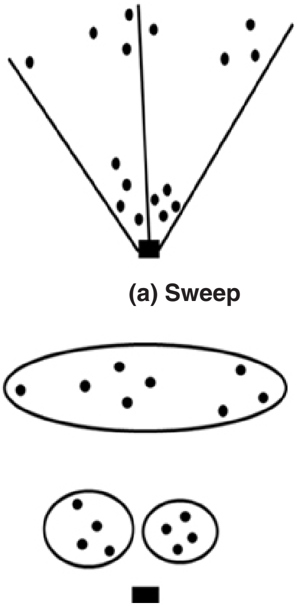
- Set C = 1.
- Start sweeping all customers by increasing polar angle and assign the customers to the current cluster.
- Stop the sweep when adding the next customer would violate the maximum vehicle capacity.
- Set C = C + 1.
- Create a new cluster by resuming the sweep where the last one left off.
- Repeat Steps 2–5, until all customers have been included into clusters.

Figure 2 shows forward sweep cluster creation starts from 0^0 and goes in anti clockwise direction. Algorithm 3 shows the sweep steps.

Sweep Nearest Algorithm

The Sweep algorithm clusters the nodes solely by polar angle. If the nodes are widely separated but have less angular difference, they may be grouped in the same cluster. This reduces the optimality of the solution cluster. To resolve this problem, an algorithm was proposed named Sweep Nearest algorithm (SN)[15], which combines the classical Sweep and the Nearest Neighbor algorithm. SN first assigns a vehicle to the customer with the smallest polar angle among the remaining customers and then finds the nearest stop to those already assigned and then inserts that customer.

Figure 3 compares Sweep and Sweep Nearest clustering. Fig. 3(a) shows the clustering of Sweep algorithm where distant nodes are inserted into the same cluster due to their polar angle. On the other hand, closest nodes belong to the same cluster in Sweep Nearest as shown in Fig. 3(b) and produces better cluster. The procedure of Sweep Nearest algorithm is shown in Algorithm 4.



(a) Sweep

(b) Sweep Nearest

Fig. 3: Comparison between Sweep and Sweep Nearest clustering

Algorithm 4: Sweep Nearest
Initialization

- Sort the customer according to their increasing order of polar angles. Insertion order is Sk, Sk+1, Sk+2, … Sn, S1, S2…Sk-1 where Sk is denoted as customer k. Set k=1.

Clustering

- Set C = 1.
- If C = 1, assign Skto the vehicle. Else assign

a customer among the remaining with the smallest angle.

- Select the nearest customer from all the nodes of the current route and insert it into the route. Continue inserting such nearest customer into the cluster until vehicle capacity is not violated.
- Set C = C + 1.
- Repeat Steps 2-4 until all the customers have been visited.
- If k<n, set k = k + 1 and got to Step 1. Else stop.

Sweep Reference Point Algorithm

In the Sweep and Sweep Nearest, the depot is used as a reference point for calculating the polar angle. Changing the reference point alters the polar angle of nodes and the insertion sequence of nodes and causes different routes to be generated. Based on these observations, the authors of[15] proposed two different kinds of reference points rather than the depot: every node and distant point.

Every Node as Reference Point

Polar angle of the customers is calculated with respect to the other customers except depot. The formula is,

$$\theta = \tan^{-1}\{(y_i - y_r) / (x_i - x_r)\} \qquad ...(5)$$

where

x_i, y_i = X, Y co-ordinates of customer i.

x_r, y_r = X, Y co-ordinates of the reference node.

Distant Point as Reference Point

The distant reference points are denoted as x^+, x^-, y^+, y^-. They are calculated using the following formula,

$$C_x = \frac{max_{\forall i}(x_i) + min_{\forall i}(x_i)}{2},$$

$$C_y = \frac{max_{\forall i}(y_i) + min_{\forall i}(y_i)}{2}$$

$$\delta_x = max_{\forall i}(x_i) - min_{\forall i}(x_i),$$

$$\delta_y = max_{\forall i}(y_i) - min_{\forall i}(y_i)$$

reference $x^+ = (C_x, C_y) + (K^*\delta_x, 0) = (C_x + K^*\delta_x, C_y$

reference $x^- = (C_x, C_y) - (K^*\delta_x), 0) = (C_x - K^*\delta_x, C_y)$

reference $y^+ = (C_x, C_y) + (0, K^*\delta_y) = (C_x, C_y + K^*\delta_x)$

reference $y^- = (C_x, C_y) - (0, K^*\delta_y) = (C_x, C_y - K^*\delta_x)$

Here, $max_{\forall i}(x_i)$ is the maximum value of X co-ordinate of all the clusters and $min_{\forall i}(x_i)$ is for the minimum value. $max_{\forall i}(y_i)$ and $min_{\forall i}(x_i)$ are for the Y co-ordinate. The clustering of nodes is same as Sweep or Sweep Nearest. K is a constant defined by the authors and it's called distant level parameter and in this study it is K = 5. Clustering in Sweep Reference Point algorithm is as same as Sweep or Sweep Nearest algorithm.

Fisher and Jaikumar Algorithm

This algorithm[11] solves a Generalized Assignment Problem (GAP) to form the clusters. In GAP, there are a number of agents and a number of tasks. This problem is a generalization of the assignment problem in which both tasks and agents have a size. Moreover, the size of each task might vary from one agent to the other. Any agent can be assigned to perform any task, incurring some cost and profit that may vary depending on the agent-task assignment. Moreover, each agent has a budget and the sum of the costs of tasks assigned to it cannot exceed this budget. It is required to find an assignment in which all agents do not exceed their budget and total profit of the assignment is maximized. When GAP is applied for CVRP, the vehicles are considered as agents, vehicle capacity is agent's budget, tasks are customers and profit is the optimal route cost. The assignment of customers to vehicles is done in such a way that vehicle capacity is not violated and the travelled distance is as less as possible.

At first, in Fihser and Jaikumar algorithm, the customers are divided into K cones where K is the number of total vehicles. Seed customers are selected from the cones, based on some criteria such as customers with maximum demand or most distant customer from origin. Fig. 4 shows that the customers (N = 15) are divided into four cones (K = 4). If the smallest angle is 200 and the largest angle is 3400 then each cone is of 800. From each cone, the farthest node from the origin is chosen as the seed customer. After seed selection, insertion cost is calculated which is the cost of

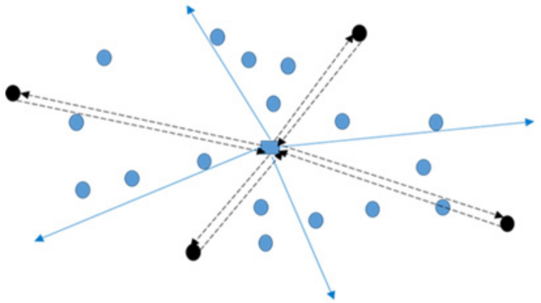


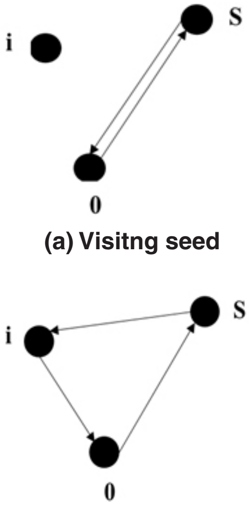
Fig. 4: Seed customer selection in Fisher and Jaikumar

inserting customer within the route going back and forth between seed customer and depot. Then the customers are assigned to vehicles according to the increasing order of insertion cost.

Figure 5 demonstrates visiting seed and a customer where 0 denotes the depot, S is seed customer and i is the customer being considered for insertion. Fig. 5(a) shows the back and forth route of seed customer and depot. Fig. 5(b) shows the route if customer i is visited in the way of returning from seed customer to depot. From Fig. 5(a), the travelled distance dS of visiting the seed customer and returning to depot is,

$$d_S = d_{0S} + d_{S0} = 2d_S \qquad ...(6)$$

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



(a) Visitng seed

(b) Visiting customer with Seed

Fig. 5: Demonstration of visiting seed alone and with a customer

$$d_i = d_{Si} + d_{io} \qquad ...(7)$$

Substituting Eq. (6) from Eq. (7), insertion cost of customer i is

$$C_{ik}= d_i – d_S= d_{Si} + d_{i0} – d_S \qquad ...(8)$$

The authors of[11] pointed some attractive attributes of the algorithm. First, the heuristic is always find a feasible solution if one exists due to the characteristics of GAP. Second, when the GAP is solved, the algorithm is considering the impact of a customer assigned to a vehicle on every other possible assignment. This avoids a problem faced by sequential assignment or limited adjustment heuristics that is unknowingly making initial assignment which may lead to expensive route generation. Third, the method can be easily adapted to accommodate the constraints of CVRP. Finally, it lets to select seed customer and thus it helps in generating different route set. The procedure of Fisher and Jaikumar algorithm is shown in Algorithm 5.

Algorithm 5: Fisher and Jaikumar

Seed Selection

- Divide the total angular planes by the number of vehicle and obtain cones equal to the number of vehicles.
- Choose a customer from each cone with maximum demand or farthest from the origin and make that seed customer.
- Assign a vehicle to each of the seed customer.

Insertion Cost Calculation

Calculate insertion cost of each customer with respect to each seed using Eq. (8).

Assignment of Customers

The customers are assigned to the vehicles according to their increasing order of insertion cost while maintaining vehicle capacity constraint.

Optimum Route Generation

The route generation stage is aimed to optimize searching method to find the optimal solution that represents the shortest path between all nodes in each cluster generated by the clustering algorithm. In this stage each cluster is an individual traveling salesman problem (TSP). In this study, three prominent methods are considered for route optimization: Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Velocity Tentative Particle Swarm Optimization (VTPSO). GA is a prominent and pioneer optimization method. ACO is the prominent Swarm Intelligence (SI) based method and is pioneer in solving TSP. On the other hand, Velocity Tentative Particle Swarm Optimization (VTPSO) is recently developed method extending PSO of continuous optimization. Brief description of the methods to optimize route (i.e., to solve TSP) are given to make the paper self-contained.

Genetic Algorithm (GA)

GA[12] 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. To produce new tour from existing tours, Enhanced Edge Recombination cross over method is used in this study. The positions of two nodes are interchanged for mutation operation.

Ant Colony Optimization (ACO)

ACO is inspired from ants' foraging behavior and is the prominent method for solving TSP[13, 16]. 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 consider 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. Finally, all the ants follow the same route after certain iteration. The detail description of ACO is available in[19].

Velocity Tentative PSO (VTPSO)

VTPSO[14] is the most recent SI based method extending Particle Swarm Optimization (PSO) to solve TSP. In PSO, a particle represents a feasible solution in multi-dimensional search space. In each iteration, each particle measure its velocity considering its own past best position and the best position encountered by the whole swarm. VTPSO calculates velocity as Swap Sequence (SS) to alter a TSP solution to new one but apply the SS in a different and optimal way. A SS is a collection of Swap Operators (SOs) and each SO is a pair of indexes to alter. On the other hand, VTPSO considers the calculated velocity SS as tentative velocity and conceives a measure called partial search (PS) to apply calculated SS to update particle's position (i.e., TSP tour). VTPSO measures tours with portions of SS and conceives comparatively better new tour with a portion or full tentative SS. The algorithm is described in[14].

Experimental Studies

This section first describes the benchmark problems and experimental setup for conducting the experiment. Then it compares outcomes of different methods.

Benchmark Data and Experimental Setup

In this study, a suite of benchmark problem named Augerat et al. (A-VRP) has been chosen[17]. Each data set includes the number of customers, number of vehicles available and each vehicle capacity. A customer is represented as a two dimensional coordinate and individual customer demand is also given. In A-VRP, number of customer varies from 32 to 80, total demand varies from 407 to 932, and number of vehicle varies from 5 to 10. The capacity of individual vehicle is 100 for all the problems. The original data set is modified here to make the coordinate of the depot as (0, 0).

We strongly relied on an experimental methodology for configuring the route optimization algorithms. For the fair comparison, the number of iteration was set to 100 for the algorithms. The population size was 50 for GA and VTPSO. The number of ants in ACO was equal to the number of nodes in a cluster as it desired. In ACO, alpha and beta were set to 1 and 3, respectively. The selected parameters are not optimal values, but considered for simplicity as well as for fairness in observation.

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.

Experimental Results and Analysis

This section presents experimental results for different clustering and route optimization methods. Route optimization is performed with GA, ACO and VTPSO for each clustering method. Table 1 shows the initial route cost and optimized route cost of Savings Series (SVS) and Savings Parallel (SVP) for benchmark problems. Table 2 is for optimized CVRP route cost of Sweep algorithm (SW), Sweep Nearest algorithm (SN), Sweep Reference Point algorithm using every stop (SRE), Sweep with Reference with Distant point (SRD) and Fisher and Jaikumar algorithm (FJ). From the tables, it can be observed that for the same instances and using same optimization technique, the route costs are different due to different approach of cluster/route generation. The best route found for each instance is shown as bold. On the other hand, in few cases a method requires an additional vehicle to cover all the nodes which are marked with a along with CVRP cost. It is observed from Table 1 that SVP is better than SVS. Without optimization, SVP is shown better CVRP cost than SVP for all the cases except n53-k7. In few cases outperformance of SVP is significant. For n33-k5 problem, as an example, SVS achieved CVRP cost 957 and SVP achieved 842. However, SVP involves more computation than SVS; SVP includes customer insertion and route merging.

Table 1 also presents effect of vehicle route optimization on both SVS and SVP. In general, optimization does not incur with Savings since it looks on overall CVRP cost while generating solutions. For better understanding, '-' indicates optimization is not found effective for the instances. From the table it is found that ACO is unable to improve CVRP solution for any cases of SVP and SVS. GA is found to improve solutions 22 and 16 cases for SVS and SVP, respectively. On the other hand, VTPSO improved all 27 cases for SVS and 18 cases for SVP. 22 and all 27 cases. At a glance, solving CVRP with Savings, SVP + VTPSO is the best method. Finally, the interesting observation from the table is that there is a scope to improve Savings solution through TSP optimization methods.

Table 1: CVRP cost of Savings algorithms (Series and Parallel) for A-VRP benchmark problems. '-' indicates optimization is not found effective for the problems. . [a] with cost indicates an additional vehicle was required to solve the problem.

SL.	Prob.	Without Optimization		Optimization with					
				GA		ACO		VTPSO	
		SVS	SVP	SVS	SVP	SVS	SVP	SVS	SVP
1	n32-k5	957	842	937	829	-	-	927	829
2	n33-k5	769	716	759	711	-	-	759	711
3	n33-k6	926	774[a]	916	-	-	-	916	-
4	n34-k5	886	809[a]	862	-	-	-	862	-
5	n36-k5	994	835	973	806	-	-	973	806
6	n37-k5	898	705	838	704	-	-	833	693
7	n37-k6	1061	977	1048	975	-	-	1048	975
8	n38-k5	847	770[a]	830	769[a]	-	-	827	-
9	n39-k5	1031	907	1020	-	-	-	1020	-
10	n39-k6	1024	857	997	855	-	-	992	855
11	n44-k6	1155	1006	1111	1005	-	-	1108	1005
12	n45-k6	1120	997[a]	1118	992[a]	-	-	1116	992[a]
13	n45-k7	1334	1198	1313	-	-	-	1311	-
14	n46-k7	1133	939	1045	937	-	-	1038	936
15	n48-k7	1234	1110	1214	-	-	-	1214	1100
16	n53-k7	1167	1198	-	1083	-	-	1146	1083
17	n54-k7	1379	1209	1361	1192	-	-	1361	1191
18	n55-k9	1277	1109	1274	-	-	-	1266	-

19	n60-k9	1638	1408	1616	1406	-	-	1614	1406
20	n61-k9	1302[a]	1058[a]	1286[a]	-	-	-	1282a	1050[a]
21	n62-k8	1622	1368	1603	-	-	-	1600	-
22	n63-k9	1904	1682[a]	-	1680[a]	-	-	1885	1678[a]
23	n63-k10	1556	1352	-	1345	-	-	1548	1342
24	n64-k9	1763	1489[a]	-	1488[a]	-	-	1701	1482[a]
25	n65-k9	1519	1266[a]	1500	-	-	-	1492	-
26	n69-k9	1312	1192	1195	-	-	-	1188	-
27	n80-k10	1967	1840	-	-	-	-	1955	1836
Best Count		0	7	0	10	0	0	1	18
Improvement due to optimization				22	16	0	0	27	18

Table 2. CVRP cost for different clustering and optimization methods for A-VRP benchmark problems. [a] with cost indicates an additional vehicle was required to solve the problem

SL	Problem	GA					ACO					VTPSO				
		SW	SN	SRE	SRD	FJ	SW	SN	SRE	SRD	FJ	SW	SN	SRE	SRD	FJ
1	n32-k5	882	815	814	827	886	897	822	822	837	894	882	814	814	826	886
2	n33-k5	791	709	712	700	705	802	707	712	699	705	788	714	712	697	738
3	n33-k6	874[a]	765	744	743	854	877[a]	767	748	748	876	874[a]	764	743	743	881
4	n34-k5	826	858	819	824	870	852	860	834	827	871	826	858	819	824	793
5	n36-k5	949	830	869	850	844	965	850	876	856	858	942	827	861	841	955
6	n37-k5	822	815	852	803	736	837	852	804	809	753	825	812	829	808	847
7	n37-k6	1141[a]	968	972	1000	1016	1141[a]	978	978	978	1041	1131[a]	966	969	1000	1016
8	n38-k5	876[a]	865	801	772	785	907a	963	802	779	781	874a	854	801	765	968
9	n39-k5	881	842	855	912	900	918	865	875	886	923	877	842	854	912	977
10	n39-k6	997	915	896	893	949	997	925	906	884	974	991	915	896	893	980
11	n44-k6	1165[a]	1031	1041	992	1162	1230[a]	1031	1031	996	1184	1164[a]	1027	1031	991	1113
12	n45-k6	1115[a]	1097[a]	981[a]	1023[a]	1034	1140[a]	1020[a]	1028[a]	1028[a]	1045	1115[a]	1029[a]	976[a]	1023[a]	1199
13	n45-k7	1344	1229	1208	1236	1199	1364	1233	1210	1273	1202	1343	1223	1206	1236	1307
14	n46-k7	977	1004	1016	1045	1040	1010	1015	1016	1037	1035	975	993	1016	1045	1045
15	n48-k7	1159	1205	1135	1200	1186	1166	1211	1150	1208	1183	1152	1200	1133	1194	1253
16	n53-k7	1183[a]	1213[a]	1169	1157	1257	1211[a]	1234[a]	1154	1182	1315	1174[a]	1266[a]	1145	1148	1173
17	n54-k7	1380[a]	1324	1198	1272	1200	1374[a]	1330	1232	1277	1203	1361[a]	1323	1125	1270	1399
18	n55-k9	1201	1143	1131	1121	1299	1215	1152	1137	1128	1385	1201	1143	1197[a]	1121	1434
10	n60-k9	1512	1484	1470	1473	1756	1562	1498	1488	1454	1756	1512	1484	1464	1473	1608
20	n61-k9	1225[a]	1169[a]	1226[a]	1160[a]	1141[a]	1238[a]	1210[a]	1128[a]	1151[a]	1154[a]	1219[a]	1169[a]	1173[a]	1145[a]	1137
21	n62-k8	1529	1417	1380	1386	1664	1554	1426	1407	1385	1668	1527	1414	1371	1386	1664
22	n63-k9	1824[a]	1758[a]	1775[a]	1836[a]	1792	1856[a]	1784[a]	1844[a]	1861[a]	1806	1823[a]	1758[a]	1772[a]	1835[a]	1959
23	n63-k10	1482[a]	1450	1428	1426	1499	1532[a]	1461	1430	1430	1499	1481[a]	1440	1426	1426	1581
24	n64-k9	1607[a]	1601[a]	1569	1653[a]	1613	1628[a]	1607[a]	1554	1673[a]	1603	1598[a]	1601[a]	1547	1601[a]	1659
25	n65-k9	1374[a]	1351[a]	1241	1394[a]	1359	1399[a]	1356[a]	1255	1353[a]	1351	1373[a]	1351[a]	1242	1329[a]	1533
26	n69-k9	1254[a]	1265	1246	1295	1323	1280[a]	1272	1295	1299	1367	1254[a]	1257	1236	1305	1393
27	n80-k10	2139[a]	1912	1936	1986	2042	2195[a]	1916	1953	2028	2099	2136[a]	1901	1884	1975	2186
Best Count		0	2	2	3	2	0	0	1	2	0	1	5	11	6	1
Opt. Alg. Best Count				9				3							21	

Results presented in Table 2 for SW is for both forward and backward where anti-clockwise and clockwise sweep have been applied to form clusters. In SW, clusters are created based on solely customer's angular position. One problem with polar angle clustering is customers separated widely may be inserted into the same cluster due to less angle difference. For this reason, Sweep Nearest (SN) was introduced which considers both angle and distance between customers to generate routes and thus produces better solution than SW. But in SN, for each insertion, distance between the clustered customers and remaining customers are computed for finding the nearest customer to insert. Also SN considers each customer as starting node and creates cluster set equal to the number of customers. As an example of n33-k5 problem, SN generates 33 clusters sets and selects the clusters set with the least CVRP cost. Thus SN requires more computation and iteration than SW and quite inefficient for large problem.

Results in Table 2 for Sweep Reference Point considered two different kind of reference points rather than depot; every stop (SRE) and distant points (SRD). It is notable that reference point is depot in SW and SN. For a particular optimization method (e.g., GA) the difference in outcome among SW, SN, SRE and SRD are due to reference point alteration. For instance n33-k5 problem, when the reference point is depot the CVRP costs after optimization with GA are 791 and 709 for SW and SN, respectively. But, when reference point changes, the cost reduced to 712 and 700 for GA for SRE and SRD, respectively. Similar effect for ACO and VTPSO are also observed. Various reference node increases the optimality of the clusters for many instances. SRD works with 4 reference points where SRE considers every node except customers. Therefore, SRE is shown better performance than SW, SN and SRD. With VTPSO optimization, SRE+VTPSO is shown best CVRP cost for 13 cases out of 27 cases.

From the Table 2 it is observed that Fisher and Jaikumar (FJ) with GA is shown to achieve best CVRP solutions for three cases n37-k5, n45-k7 and n61-k9. FJ selects distant customers as seed. Therefore, it considers farthest customers from depot for insertion at first and nearest ones later. The distant customers influences the cost of the clusters more than the nearest. Because the customers closer to depot can be inserted into any clusters and it will have less impact on the cost than the distant customers' insertion. It performs better when the customers are evenly distributed around the depot. It creates all the clusters in parallel, considers the insertion of a customer in all the cluster and inserts into the best suitable one. But customer density may vary in different regions; thus seed selection influences optimality of the routes. In some angular area customers may not even exists. When the customers are densely located in a particular area, more seeds get selected from that area comparing with the other areas. SW and SN outperform FJ for most of the instances. But FJ also produces feasible clusters (where cluster number does not exceed the vehicles provided) and involves less computation time.

Considering results of Table 1 and Table 2, after implementing different optimization techniques, the CVRP costs do not vary for constructive algorithm as much as for clustering algorithm. Also Savings algorithm (Table 1) is giving same result for different optimization algorithms. On the other hand, for clustering algorithm GA is performing better than ACO; and VTPSO is outperforming both GA and ACO. Among 27 instances, VTPSO is giving best result for 22 cases.

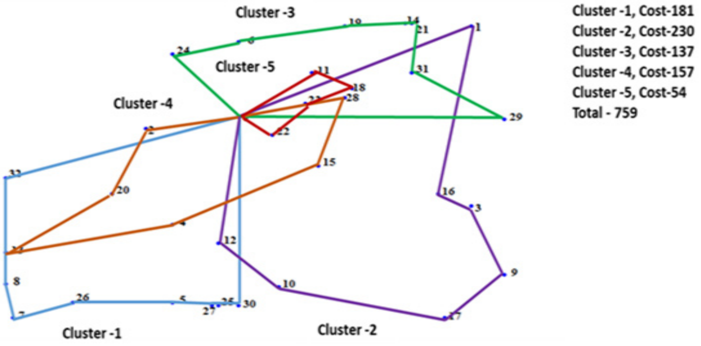


Fig. 6: CVRP solution of n33-k5 with GA on SVS

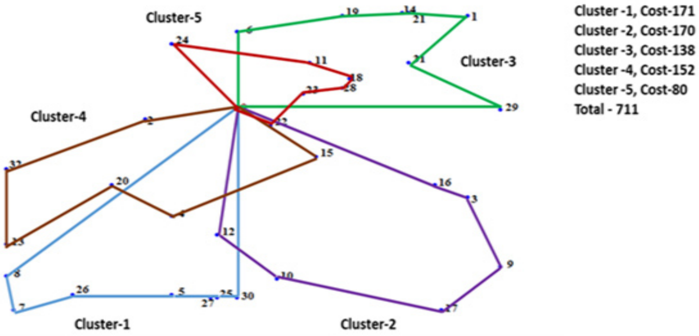


Fig. 7: CVRP solution of n33-k5 with GA on SVP

Figures 6 to 11 show graphical representation of final CVRP solutions with route optimization with GA for SVS, SVP, SW, SN, SRD and FJ for sample problem n33-k5. Fig. 6 and Fig. 7 demonstrate the CVRP solutions for SVS and SVP, respectively. Common feature of SVS and SVP is that both contains intersections clusters. On the other hand, clusters are varied in the two method with different nodes. Node 24 is in cluster 3 in SVS and in Cluster 5 in SVP. Total CVRP costs for SVS and SVP are 759 and 711, respectively.

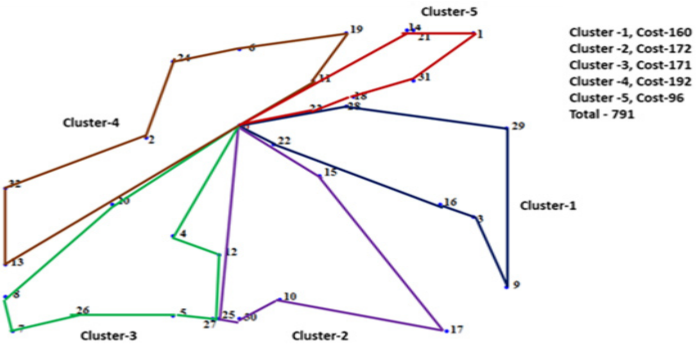


Fig. 8: CVRP solution of n33-k5 with GA on SW

Figure 8 shows the CVRP solution for SW+GA. There is no intersection in clusters in the route. SW clustering starts from a particular angle and advances only either forward or backward; thus there is no scope of intersection. Although routes of individual looks nice but CVRP cost is relatively high and is 791.

Figure 9 and Fig. 10 demonstrate the CVRP solutions for two SW variants SN and SRD,

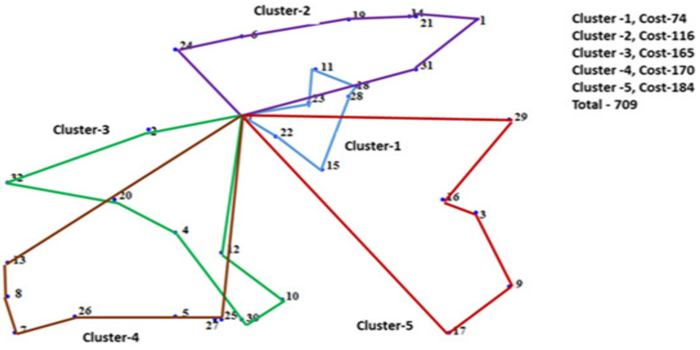


Fig. 9: CVRP solution of n33-k5 with GA on SN

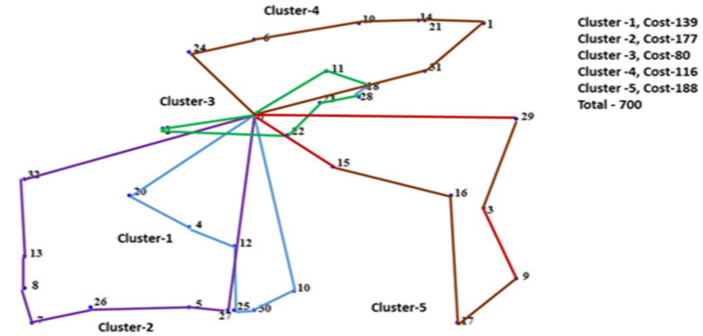


Fig. 10: CVRP solution of n33-k5 with GA on SRD

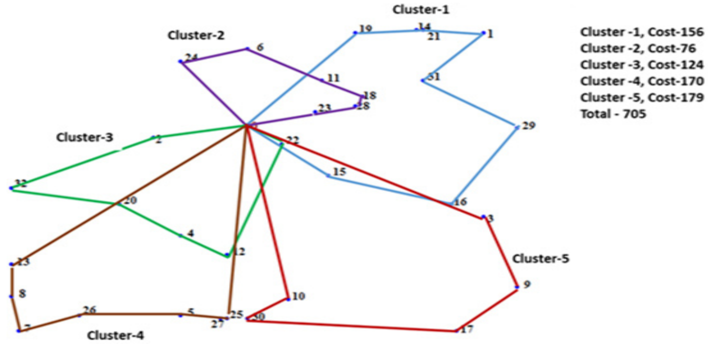
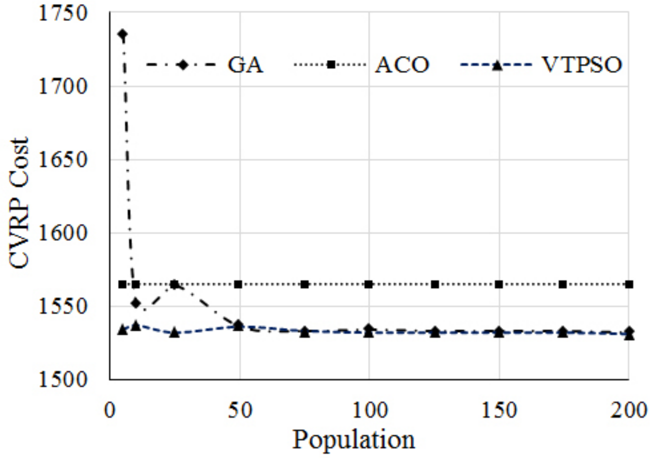


Fig. 11: CVRP solution of n33-k5 with GA on FJ

respectively. Common feature of the methods is cluster intersections. In the methods cluster formation varies due to reference point's variation. On the other hand, clusters are varied in the two method with different nodes. The CVRP cost of the methods are promising and achieved cost of 700 by SRD is the best among the methods with GA. Finally, Fig. 11 shows the solution with FJ which is similar to SRD. Hence, CVRP cost of FJ is also competitive to SRD and achieved value is 705.

There is an effect of population size on performance in any population based method. For better understanding, the effect of population size is investigated for GA and VTPSO varying population size from 5 to 200. The population size in ACO was the number nodes in a particular cluster. While varying population, the number of generation was fixed at 100. Fig. 12 shows CVRP route costs on



SW based clusters of n62-k8 problem. From the figure it is observed that performance of GA is highly dependent on population size. It showed very bad CVRP cost for very small population (e.g., 5) and became steady for population with 50 or more. VTPSO also found population dependent but performed relatively better for small population. At a glance, VTPSO outperformed both GA and ACO.

Conclusion

CVRP is a popular combinatorial optimization problem and interest grows in recent years to solve it new ways. Two popular ways of solving CVRP are constructive and clustering methods. Constructive method based algorithms create routes and minimize the cost at the same time. On the other hand, clustering algorithms require two phase to get an optimal solution. Constructive algorithms do not consider alternate solutions. Also constructive method performs better than clustering in most of the instances. Finally, GA, ACO and VTPSO are applied to generate optimal route with the both constructive and clustering algorithms' outcome. Although optimization technique generally not consider constructive approach, this study revealed that there is a scope to improve Savings solution through TSP optimization methods. Among the three route optimization methods, VTPSO is shown the best and performed well with SVP and SRE.

Disclosure Statement

Authors declared that there is no conflict of interest.

References

1 G. B. Dantzig and J. H. Ramser, "The Track Dispatching Problem", INFORMS, *Management Science*, vol. **6**, no. 1 pp. 80-91, October, 1959.

2 Christofides N, Mingozzi A, Toth P,"The vehicle routing problem", In: Christofides N, Mingozzi A, Toth P, Sandi C, editors.

3 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.

4 Jens Lysgaard, "Clarke & Wright's Savings Algorithm", Department of Management Science and Logistics, the Aarhus School of Business, 1997.

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

6 T. Doyuran, B. Çatay, "A robust enhancement to the Clarke-Wright savings algorithm", Technical notes, Faculty of Engineering and Natural Sciences, Sabanci University, Tuzla, Istanbul, Turkey, 2011.

7 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.

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

9 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.

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

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

12 D. E. Goldberg, "Genetic Algorithm in Search, Optimization and Machine Learning," New York: Addison – Wesley, 1989.

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

14 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.

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

16 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.

17 Large Capacitated Vehicle Routing Problem Instances. Available: http://neo.lcc.uma.es/vrp/vrp-instances/