HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
---------------o0o---------------

SEMINAR 1

Particle Swarm Optimization for Capacitated Vehicle Routing Problem

Lecturers **:** Prof. Michel Toulouse

Group member : Hoang Hong Ha - 20212306M

HANOI, May 2021

Table of Contents

I. Introduction

The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem which asks "What is the optimal set of routes for a fleet of vehicles to traverse to deliver to a given set of customers?". It generalizes the well-known traveling salesman problem (TSP)[1]

The VRP was first proposed by Dantzig and Ramser (1959) and has been widely studied. According to L. Guerra et al. (2007) and S. Masrom et al. (2010). Often, the context is that of delivering goods located at a central depot to customers who have placed orders for such goods. VRP will solve the problem in which a set of routes for a fleet of delivery vehicles based at one or several depots must be determined for several customers. The main objective of VRP is to serve customer demands by a minimum cost vehicle routes originating and terminating in a depot. In 1964, Clarke and Wright improved on Dantzig and Ramser's approach using an effective greedy algorithm called the savings algorithm. Determining the optimal solution to VRP is NP-hard, so the size of problems that can be solved, optimally, using mathematical programming or combinatorial optimization may be limited. Therefore, commercial solvers tend to use heuristics due to the size and frequency of real-world VRPs they need to solve. Several variations of the VRP exist to adapt to various practical characteristics and constraints such as Multiple Depot VRP (MDVRP), Split Delivery VRP (SDVRP), and Dynamic VRP (DVRP). If a constraint is given on the capacity of every vehicle, the problem is known as capacitated vehicle routing problem (CVRP).

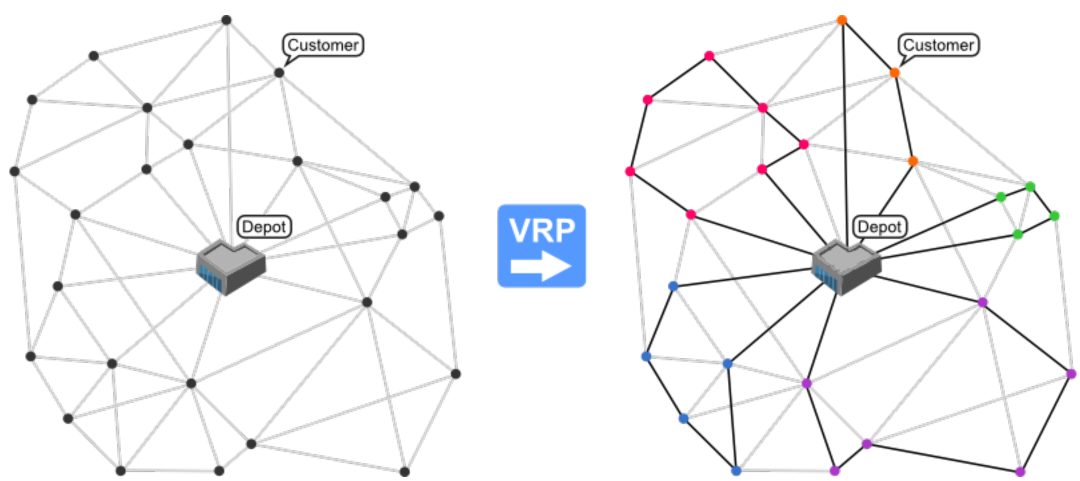

Figure 1. VRP with One Depot and 25 Customers

II. Problem: Capacitated vehicle routing problem

Capacitated Vehicle Routing Problem (CVRP) as defined by J. F. Cordeau (2002) and J. Lysgaard (2004) is a set of N customers with determined demands which must be served from a common depot by a fleet of delivery vehicles that has a constraint on their capacity. After completing a tour from the depot and serving some customers along its route is the summation of the Euclidean distance between each pair of nodes that the vehicle visits (see Figure 2).

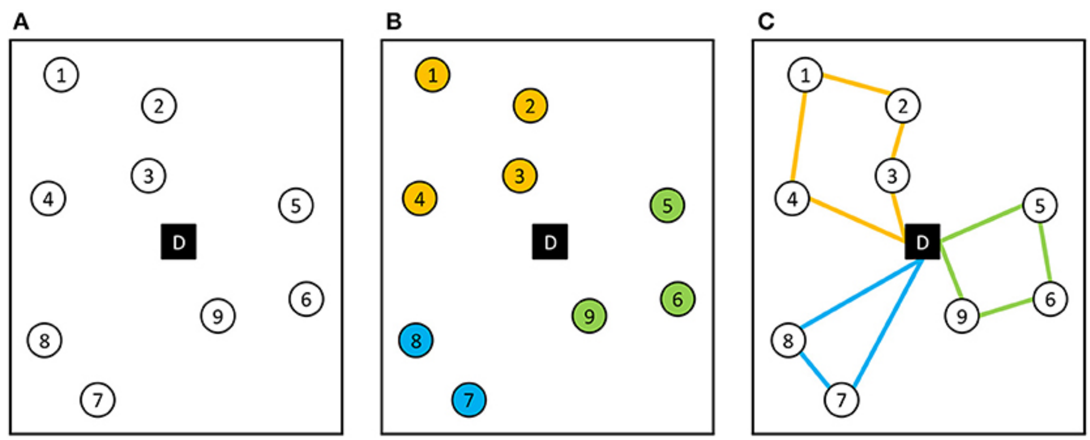

Figure 2. CVRP optimizes routes from one depot to 9 geographically scattered customers.

The CVRP aims are to find a set of minimum total cost routes for a fleet of capacitated vehicles based at a single depot, to serve a set of customers under the following constraints:

- Each route begins and ends at the depot.
- Each customer is visited exactly once.
- The total demand for each route does not exceed the capacity of the vehicle.

2.1. The objective function

Suppose that depot is 0 and the customers should be served by available vehicles. The demand of the customer C_i is q_i, the capacity of vehicle k is Q_k and the maximum travel distance by vehicle k is D_k. The mathematical model of CVRP is described as follows[2]:

The number of customers is determined by N, the number of vehicles is K, and the traveling cost by vehicle k from customer i to j is shown as C_{ij}^k and d_{ij}^k is the traveled distance between customer i to j.

The binary variable X_{ij}^k has a value of 1 if the arc from node *i* to node *j* is in the optimal route and is driven by vehicle *k*.

$$X_{ij}^k \in \{0,1\} \qquad \forall k \in \{1, \dots, K\},\ \ i, j \in \{1, \dots, N\}$$

Whereby, there is no travel from a node to itself:

$$X_{ij}^k = 0 \qquad \forall k \in \{1, \dots, K\},\ \ i, j \in \{1, \dots, N\}$$

We have an objective function:

$$\sum_{k=0}^{K} \sum_{i=0}^{N} \sum_{j=0}^{N} C_{ij}^k X_{ij}^k \quad (1)$$

The objective function of Equation 1 is to minimize the total cost for all vehicles that are the sum of the travel distance of vehicles in the problem space.

$$\sum_{k=0}^{K} \sum_{i=0}^{N} X_{ij}^k = 1, j = 1, 2, \dots, N \quad (2)$$

$$\sum_{k=0}^{K} \sum_{j=0}^{N} X_{ij}^k = 1, i = 1, 2, \dots, N \quad (3)$$

Constraints Equation 2 and Equation 3 ensure that each customer is served exactly once.

$$\sum_{i=0}^{N} X_{it}^k - \sum_{j=0}^{N} X_{tj}^k = 0, k = 1,2, \dots, K; t = 1,2, \dots N \quad (4)$$

Constraint Equation 4 ensures the connectivity of the route.

$$\sum_{i=0}^{N} \sum_{j=0}^{N} d_{ij}^k X_{ij}^k \leq D_k, k = 1, 2, \dots, K \quad (5)$$

Constraint Equation 5 shows that the total length of each route has a limit.

$$\sum_{j=0}^{N} q_j \left(\sum_{i=0}^{N} X_{ij}^{k} \right) \leq Q_k, k = 1,2, \dots, K \quad (6)$$

Constraint Equation 6 shows that the total demand of any route must not exceed the capacity of the vehicle

$$\sum_{j=0}^{N} X_{0j}^{k} \leq 1, k = 1,2, \dots, K \quad (7)$$

$$\sum_{i=0}^{N} X_{i0}^{k} \leq 1, k = 1,2, \dots, K \quad (8)$$

Constraint Equation 7 and Equation 8 ensure that each vehicle is used no more than once.

$$X_{ij}^{k} \in \{0,1\};\ i,j\ = 0,1, \dots, N;\ k = 1,2, \dots K \quad (9)$$

Constraint Equation 9 ensures that the variable only takes the integer 0 or 1.

III. Particle Swarm Optimization

In computational science, particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with about a given measure of quality. It solves a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search space according to simple mathematical formula over the particle's position and velocity. Each particle's movement is influenced by its local best-known position but is also guided toward the best-known positions in the search space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. Also, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found.[3]

3.1. PSO Algorithm

Particle Swarm Optimization (PSO) is a global optimization technique. It is originally attributed to Kennedy and Eberhart (1995). A swarm consists of a set of particles each particle represents a potential solution. Suppose that each solution is represented as a point in N-Dimensional space that each point or particle has an initial velocity, particles move through solution space, and after each time step, particles are evaluated according to some fitness criterion. They are accelerated towards particles with the best fitness value within their communication group. This property of PSO helps particles escape from local optimal solutions. Each particle has a simple memory that remembers the position of the best solution achieved by itself, this value is called personal best (pbest) and the position of the best solution obtained so far by any particle in the neighborhood of that particle, known as global best (gbest). The basic concept of PSO lies in accelerating each particle towards its pbest and the gbest locations, with a random weighted acceleration at each time step.

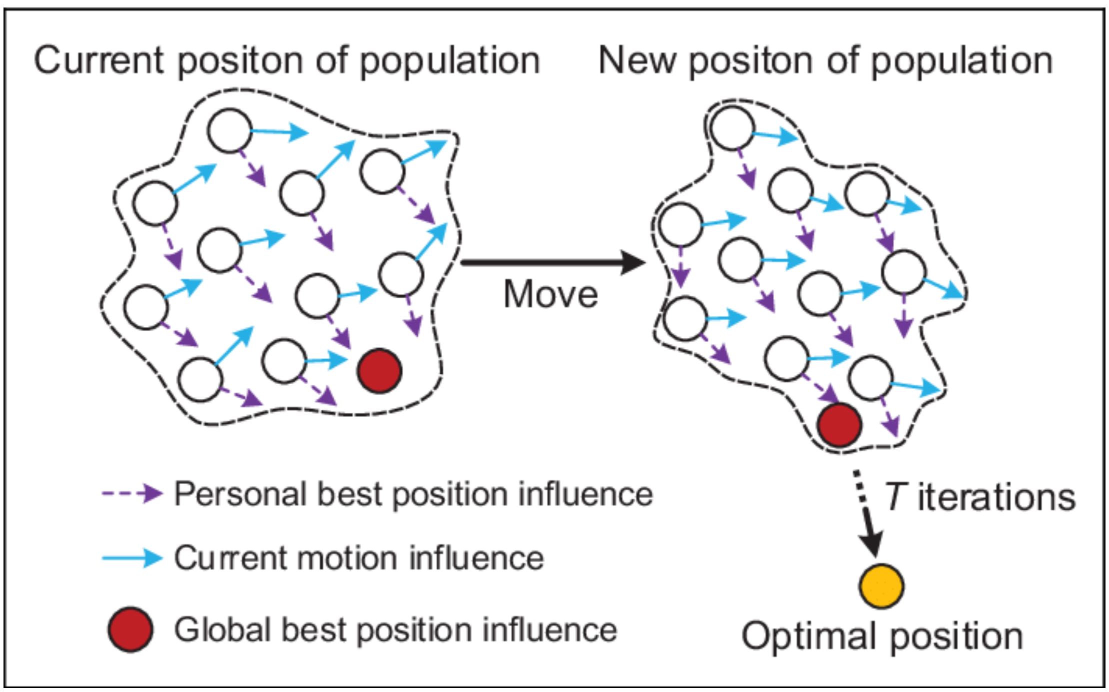

Figure 3. The Particle Swarm Optimization.

A simple example of a foraging process of a flock of birds. The foraging space is now the entire three-dimensional space that we live in. At the beginning of the search the whole flock flies in a certain direction, it can be very random. However, after a while of searching, some individuals in the herd began to find a place to contain food. Depending on the amount of food just searched, the individual sends a signal to other individuals searching in the vicinity. This signal propagates throughout the population. Based on the information received, each individual will adjust their flight direction and speed in the direction of where there is the most food. Such communication is often viewed as a phenotype of herd intelligence. This mechanism helps the whole flock of birds to find out where there is the most food in the extremely large search space.

This compromise is formalized by the following equations :

$$v_i^{t+1} = w * v_i^t + c_1 * \alpha * (p_i - x_i) + c_2 * \beta * (p_{glob} - x_i) \quad (10)$$
$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

Where:

- v_i: Particle velocity
- $w \in [0 \ldots 1]$: Inertia weight
- x_i: Current solution for particle *i*
- p_i: The current best solution for particle *i*
- p_{glob}: Global Best, the best solution from the whole swarm
- $c_1$and c_2 are so-called “acceleration” constants
- $\alpha, \beta \in [0 \ldots 1]$ respectively the weight of individual best and swarm best.

3.2. Pseudocode of PSO

```
Algo-PSO( ) {
  Initialize a population of particles ; t = 0 ;
  while (a stop criterion is not satisfied){
    for each particle x_i
      if (f(x_i) is better than p_i) (current best of particle i) then
        p_i = x_i ;
    Define p_glob as the best position found so far by any particle ;
    for each particle i {
      v_i^{t+1} = Compute_velocity(x_i^t , p_i^t , p_glob^t) ;
      x_i^{t+1} = update_position(x_i^t, v_i^t) ;
      }
  }
```

Figure 4. The pseudocode of Particle Swarm Optimization algorithm.

PSO is a metaheuristic initially designed for solving optimization problems on continuous domains.

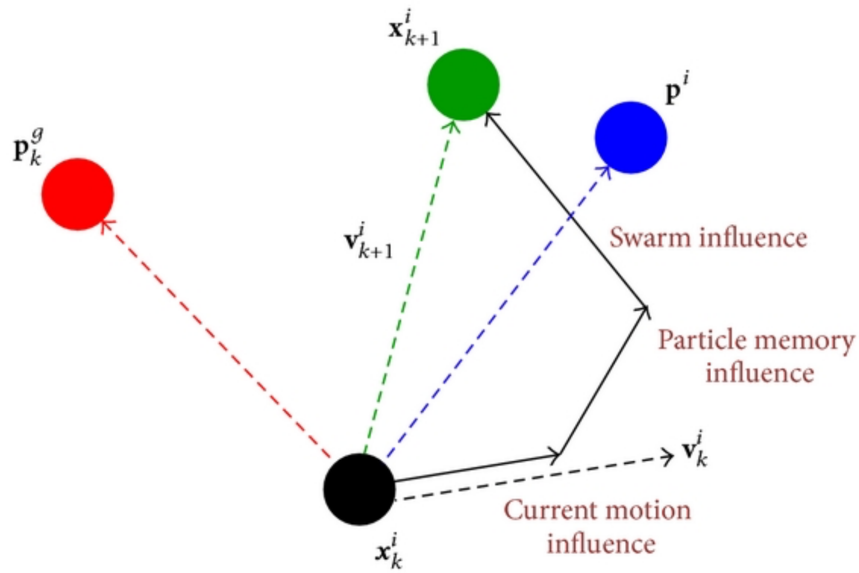

Figure 5. Illustration of velocity and position updates in PSO algorithm.

In the equation $v_i^{t+1} = w * v_i^t + c_1 * \alpha * (p_i - x_i) + c_2 * \beta * (p_{glob} - x_i)$ values can be interpreted as coordinates in a vector space and operators combining positions in the rate of change in a d-demensional space.

IV. Implementation

4.1. The datasets

The performance of the proposed algorithm has been validated on groups of benchmark problems. The benchmark problems consist of six sets as follow: seventy-four instances (sets A, B and P) from [4], eleven instances (set E) from [5], three instances (set F) from [6], five instances (set M) from [7]. All the problem instances were downloaded from the site http://vrp.atd-lab.inf.puc-rio.br. This site was created in 2014 by Ivan Lima, and by the authors of (Uchoa et al. 2017), and is maintained by Daniel Oliveira (2016-2018) and by Eduardo Queiroga (2019-present).

4.2. The parameter settings

The object PSO needs to update the speed and position of its motion, and we use the concept of Clerc and Kennedy 2002 [8]. It studied the effect of parameters φ_{1max} and φ_{2max} on v_{id}. If $(\varphi_{1max} + \varphi_{2max}) > 4$ then a constrictor coefficient, χ can be used to prevent velocity "explosion". So we have the "chi":

$$\chi = \frac{2K}{\left|2 - \varphi - \sqrt{|\varphi^2 - 4\varphi|}\right|} \qquad (11)$$

Where:

$$\varphi = \varphi_{1max} + \varphi_{2max} \qquad K \in (0, 1]$$

The parameters for PSO are set as follows:

- Kappa: $K = 1$
- Phi one: $\varphi_{1max} = 2.05$
- Phi two: $\varphi_{2max} = 2.05$
- Inertia weight: $w = \chi$
- Acceleration constants: $c_1 = \chi * \varphi_{1max}$; $c_2 = \chi * \varphi_{2max}$
- Alpha and Beta random in range (0,1)
- The number of particles: 100
- The number of iterations: 1000

V. Experiment result

We run the PSO algorithm with six sets, for each dataset we run ten times and get the best value cost, and best value cost and calculate the average value. We summarized the results given in Tables 1-9, the tables contain information on the dataset, the best value cost of this dataset, and our result for the comparison.

The first time run the algorithm, we just set the parameters $w = 1, c_1 = 2, c_2 = 1$ and the results given in Tables 1-6

Table 1 Computational results for the problem set A.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
A-n32-k5	32	100	5	1492	1492	1612	1570	47
A-n33-k5	33	100	5	1210	1210	1411	1330	73
A-n33-k6	33	100	6	1281	1281	1365	1337	31
A-n34-k5	34	100	7	1459	1459	1527	1484	28
A-n36-k5	36	100	5	1511	1511	1667	1599	54
A-n37-k5	37	100	5	1471	1471	1605	1526	47
A-n37-k6	37	100	6	1671	1671	1809	1767	49
A-n38-k5	38	100	5	1565	1565	1666	1630	38
A-n39-k5	39	100	5	1619	1619	1767	1695	50
A-n39-k6	39	100	6	1561	1561	1859	1753	111
A-n44-k6	44	100	6	1881	1881	2038	1966	60
A-n45-k6	45	100	6	2105	2105	2325	2260	85
A-n45-k7	45	100	7	2109	2109	2148	2137	15
A-n46-k7	46	100	7	2018	2018	2091	2071	28
A-n48-k7	48	100	7	2329	2329	2419	2375	36

A-n53-k7	53	100	7	2423	2423	2599	2469	65
A-n54-k7	54	100	7	2583	2583	2731	2647	50
A-n55-k9	55	100	9	2530	2530	2601	2554	25
A-n60-k9	60	100	9	2981	2981	3077	3016	33
A-n61-k9	61	100	9	2476	2476	2613	2553	53
A-n62-k8	62	100	8	3047	3047	3227	3165	62
A-n63-k9	63	100	9	3349	3349	3538	3464	64
A-n63-k10	63	100	10	2881	2881	3122	3011	77
A-n64-k9	64	100	9	2984	2984	3154	3080	72
A-n65-k9	65	100	9	3097	3097	3247	3181	51
A-n69-k9	69	100	9	3093	3093	3291	3199	75
A-n80-k10	80	100	10	4108	4108	4325	4194	84

Table 2 Computational results for the problem set B.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
B-n31-k5	31	100	5	672	924	1041	987	41
B-n34-k5	34	100	5	788	1185	1418	1323	85
B-n35-k5	35	100	5	955	1710	1925	1815	80
B-n38-k6	38	100	6	805	1469	1598	1526	49
B-n39-k5	39	100	5	549	1479	1734	1590	86
B-n41-k6	41	100	6	829	1765	1958	1848	68
B-n43-k6	43	100	6	742	1477	1621	1556	55
B-n44-k7	44	100	7	909	1752	1996	1872	82
B-n45-k5	45	100	5	751	1667	2034	1879	120
B-n45-k6	45	100	6	678	1401	1544	1469	46
B-n50-k7	50	100	7	741	1860	2164	2042	113
B-n50-k8	50	100	8	1312	2181	2315	2261	47
B-n51-k7	51	100	7	1032	2584	2815	2707	84
B-n52-k7	52	100	7	747	2076	2239	2149	53
B-n56-k7	56	100	7	707	1944	2319	2224	142
B-n57-k7	57	100	7	1153	2826	3022	2941	81
B-n57-k9	57	100	9	1598	2939	3010	2964	27
B-n63-k10	63	100	10	1496	3340	3523	3454	62
B-n64-k9	64	100	9	861	2433	2660	2553	77
B-n66-k9	66	100	9	1316	2970	3151	3100	66
B-n67-k10	67	100	10	1032	2639	2902	2775	102
B-n68-k9	68	100	9	1272	2835	3328	3191	187
B-n78-k10	78	100	10	1221	3534	3724	3639	71

Table 3 Computational results for the problem set E.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
E-n22-k4	22	6000	4	375	595	636	618	15
E-n23-k3	23	4500	3	569	949	1050	1003	42
E-n30-k3	30	4500	3	534	1002	1181	1093	71

E-n33-k4	33	8000	4	835	1283	1410	1346	47
E-n51-k5	51	160	5	521	1353	1419	1384	25
E-n76-k7	76	220	7	682	2151	2247	2214	34
E-n76-k8	76	180	8	735	2136	2261	2202	43
E-n76-k10	76	140	10	830	2238	2255	2244	6
E-n76-k14	76	100	14	1021	2276	2401	2340	43
E-n101-k8	101	200	8	817	2902	3059	2993	65
E-n101-14	101	112	14	1077	3146	3213	3184	27

Table 4 Computational results for the problem set F.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
F-n45-k4	45	2010	4	724	1843	1936	1893	31
F-n72-k4	72	30000	4	237	1060	1118	1091	20
F-n135-k7	135	2210	7	1162	5681	5875	5794	82

Table 5 Computational results for the problem set M.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
M-n101-k10	101	200	10	820	3450	3566	3510	38
M-n121-k7	121	200	7	1034	5693	5943	5806	94
M-n151-k12	151	200	12	1053	4529	4658	4598	45
M-n200-k16	200	200	16	1274	6007	6226	6119	78
M-n200-k17	200	200	17	1373	6117	6172	6143	20

Table 6 Computational results for the problem set P.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
P-n16-k8	16	35	8	450	458	475	466	6
P-n19-k2	19	160	2	212	300	336	317	14
P-n20-k2	20	160	2	216	300	365	336	22
P-n21-k2	21	160	2	211	351	379	362	10
P-n22-k2	22	160	2	216	333	397	376	23
P-n22-k8	22	3000	8	603	731	749	736	7
P-n23-k8	23	40	8	529	621	654	639	12
P-n40-k5	40	140	5	458	992	1030	1004	14
P-n45-k5	45	150	5	510	1205	1286	1245	27
P-n50-k7	50	150	7	554	1258	1339	1312	30
P-n50-k8	50	120	8	631	1323	1433	1370	36
P-n50-k10	50	100	10	696	1332	1411	1365	27

P-n51-k10	51	80	10	741	1423	1496	1455	25
P-n55-k7	55	170	7	568	1402	1488	1453	31
P-n55-k10	55	115	10	694	1469	1570	1523	36
P-n55-k15	55	70	15	989	1603	1661	1633	22
P-n60-k10	60	120	10	744	1676	1740	1717	22
P-n60-k15	60	80	15	968	1779	1863	1825	33
P-n65-k10	65	130	10	792	1927	1980	1945	19
P-n70-k10	70	135	10	827	1996	2112	2070	43
P-n76-k4	76	350	4	593	2063	2189	2152	47
P-n76-k5	76	280	5	627	2075	2196	2157	43
P-n101-k4	101	400	4	681	2969	3012	2982	16

Then, we use the concept Clerc and Kennedy with equation (11) for set value parameters of the PSO algorithm the results are given in Tables 7-12

Table 7 Computational results for the problem set A.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
A-n32-k5	32	100	5	784	1416	1686	1617	78
A-n33-k5	33	100	5	661	1199	1443	1315	75
A-n33-k6	33	100	6	742	1201	1483	1347	100
A-n34-k5	34	100	7	778	1466	1596	1523	39
A-n36-k5	36	100	5	799	1514	1705	1616	65
A-n37-k5	37	100	5	669	1368	1586	1500	69
A-n37-k6	37	100	6	949	1713	1825	1764	37
A-n38-k5	38	100	5	730	1545	1716	1649	55
A-n39-k5	39	100	5	822	1562	1838	1669	84
A-n39-k6	39	100	6	831	1642	1929	1797	91
A-n44-k6	44	100	6	937	1911	2081	2006	56
A-n45-k6	45	100	6	944	2140	2318	2238	52
A-n45-k7	45	100	7	1146	2005	2164	2091	47
A-n46-k7	46	100	7	914	1862	2175	2111	86
A-n48-k7	48	100	7	1073	2215	2456	2344	86
A-n53-k7	53	100	7	1010	2240	2549	2429	88
A-n54-k7	54	100	7	1167	2545	2812	2692	86
A-n55-k9	55	100	9	1073	2481	2684	2600	67
A-n60-k9	60	100	9	1354	2939	3240	3082	97
A-n61-k9	61	100	9	1034	2533	2684	2605	52
A-n62-k8	62	100	8	1288	3049	3225	3141	49
A-n63-k9	63	100	9	1616	3322	3617	3511	98
A-n63-k10	63	100	10	1314	2878	3111	3031	64
A-n64-k9	64	100	9	1401	2990	3210	3083	64
A-n65-k9	65	100	9	1174	3103	3278	3177	56
A-n69-k9	69	100	9	1159	3083	3368	3243	84
A-n80-k10	80	100	10	1763	4186	4409	4299	63

Table 8 Computational results for the problem set B.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
B-n31-k5	31	100	5	672	878	1123	1019	85
B-n34-k5	34	100	5	788	1239	1484	1373	77
B-n35-k5	35	100	5	955	1593	2016	1806	117
B-n38-k6	38	100	6	805	1438	1693	1599	76
B-n39-k5	39	100	5	549	1311	1722	1558	140
B-n41-k6	41	100	6	829	1763	2024	1891	77
B-n43-k6	43	100	6	742	1334	1669	1571	89
B-n44-k7	44	100	7	909	1939	2030	1982	29
B-n45-k5	45	100	5	751	1876	2083	1995	53
B-n45-k6	45	100	6	678	1456	1588	1524	36
B-n50-k7	50	100	7	741	2006	2247	2123	61
B-n50-k8	50	100	8	1312	2225	2433	2327	71
B-n51-k7	51	100	7	1032	2617	3056	2792	156
B-n52-k7	52	100	7	747	1912	2407	2224	145
B-n56-k7	56	100	7	707	2116	2439	2257	105
B-n57-k7	57	100	7	1153	2607	3177	2943	185
B-n57-k9	57	100	9	1598	2916	3103	2994	49
B-n63-k10	63	100	10	1496	3241	3632	3516	106
B-n64-k9	64	100	9	861	2448	2753	2584	99
B-n66-k9	66	100	9	1316	3005	3281	3132	82
B-n67-k10	67	100	10	1032	2823	2989	2913	55
B-n68-k9	68	100	9	1272	3169	3374	3277	56
B-n78-k10	78	100	10	1221	3362	3780	3579	111

Table 9 Computational results for the problem set E.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
E-n22-k4	22	6000	4	375	485	660	606	56
E-n23-k3	23	4500	3	569	879	1113	1017	71
E-n30-k3	30	4500	3	534	1094	1264	1191	45
E-n33-k4	33	8000	4	835	1215	1441	1332	62
E-n51-k5	51	160	5	521	1350	1482	1424	36
E-n76-k7	76	220	7	682	2099	2319	2223	62
E-n76-k8	76	180	8	735	2144	2311	2223	49
E-n76-k10	76	140	10	830	2132	2370	2268	66
E-n76-k14	76	100	14	1021	2305	2432	2369	38
E-n101-k8	101	200	8	817	2971	3112	3048	47
E-n101-14	101	112	14	1077	3048	3250	3171	59

Table 10 Computational results for the problem set F.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
F-n45-k4	45	2010	4	724	1836	2112	1948	94
F-n72-k4	72	30000	4	237	1038	1138	1088	37
F-n135-k7	135	2210	7	1162	5641	5962	5817	103

Table 11 Computational results for the problem set M.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
M-n101-k10	101	200	10	820	3445	3662	3560	78
M-n121-k7	121	200	7	1034	5690	5951	5779	82
M-n151-k12	151	200	12	1053	4532	4735	4630	66
M-n200-k16	200	200	16	1274	6045	6189	6149	45
M-n200-k17	200	200	17	1373	5994	6210	6138	59

Table 12 Computational results for the problem set P.

Problem	n	Q	K	BKS (best-known solution)	Proposed PSO			
					Best	Worst	Average	Std(standard deviation)
P-n16-k8	16	35	8	450	466	513	481	14
P-n19-k2	19	160	2	212	274	373	340	35
P-n20-k2	20	160	2	216	309	399	352	29
P-n21-k2	21	160	2	211	288	412	374	39
P-n22-k2	22	160	2	216	313	391	365	23
P-n22-k8	22	3000	8	603	674	765	724	29
P-n23-k8	23	40	8	529	592	671	634	22
P-n40-k5	40	140	5	458	941	1119	1057	46
P-n45-k5	45	150	5	510	1197	1350	1265	47
P-n50-k7	50	150	7	554	1285	1384	1328	26
P-n50-k8	50	120	8	631	1283	1454	1382	49
P-n50-k10	50	100	10	696	1313	1445	1417	38
P-n51-k10	51	80	10	741	1402	1582	1527	48
P-n55-k7	55	170	7	568	1422	1515	1470	33
P-n55-k10	55	115	10	694	1497	1612	1545	32
P-n55-k15	55	70	15	989	1577	1685	1638	40
P-n60-k10	60	120	10	744	1637	1794	1734	49
P-n60-k15	60	80	15	968	1730	1873	1822	40
P-n65-k10	65	130	10	792	1849	1996	1927	50
P-n70-k10	70	135	10	827	2045	2157	2112	39
P-n76-k4	76	350	4	593	2041	2268	2179	63
P-n76-k5	76	280	5	627	2047	2252	2173	59
P-n101-k4	101	400	4	681	2870	3085	3007	56

optimization So, we can see the algorithm doesn't give the best results. Actual results differ greatly from expected results. Further, when we compare the results given after we apply the equation (11) and the results given before, we see the results are variable but not stable and with little improvement. The comparison through column *%Dev* in the following tables.

$$\%Dev = 100 * (average\ PSO\ -\ average\ PSO\ optimize)/avarage\ PSO\ optimize$$

Table 13 Compare result of PSO after applying optimization for set A

Problem	PSO	PSO with optimize	%Dev
A-n32-k5	1570	1617	-2.91
A-n33-k5	1330	1315	1.14
A-n33-k6	1337	1347	-0.74
A-n34-k5	1484	1523	-2.56
A-n36-k5	1599	1616	-1.05
A-n37-k5	1526	1500	1.73
A-n37-k6	1767	1764	0.17
A-n38-k5	1630	1649	-1.15
A-n39-k5	1695	1669	1.56
A-n39-k6	1753	1797	-2.45
A-n44-k6	1966	2006	-1.99
A-n45-k6	2260	2238	0.98
A-n45-k7	2137	2091	2.20
A-n46-k7	2071	2111	-1.89
A-n48-k7	2375	2344	1.32
A-n53-k7	2469	2429	1.65
A-n54-k7	2647	2692	-1.67
A-n55-k9	2554	2600	-1.77
A-n60-k9	3016	3082	-2.14
A-n61-k9	2553	2605	-2.00
A-n62-k8	3165	3141	0.76
A-n63-k9	3464	3511	-1.34
A-n63-k10	3011	3031	-0.66
A-n64-k9	3080	3083	-0.10
A-n65-k9	3181	3177	0.13
A-n69-k9	3199	3243	-1.36
A-n80-k10	4194	4299	-2.44

Table 14 Compare result of PSO after applying optimization for set B

Problem	PSO	PSO with optimize	%Dev
B-n31-k5	987	1019	-3.14
B-n34-k5	1323	1373	-3.64
B-n35-k5	1815	1806	0.50
B-n38-k6	1526	1599	-4.57
B-n39-k5	1590	1558	2.05
B-n41-k6	1848	1891	-2.27

B-n43-k6	1556	1571	-0.95
B-n44-k7	1872	1982	-5.55
B-n45-k5	1879	1995	-5.81
B-n45-k6	1469	1524	-3.61
B-n50-k7	2042	2123	-3.82
B-n50-k8	2261	2327	-2.84
B-n51-k7	2707	2792	-3.04
B-n52-k7	2149	2224	-3.37
B-n56-k7	2224	2257	-1.46
B-n57-k7	2941	2943	-0.07
B-n57-k9	2964	2994	-1.00
B-n63-k10	3454	3516	-1.76
B-n64-k9	2553	2584	-1.2
B-n66-k9	3100	3132	-1.02
B-n67-k10	2775	2913	-4.74
B-n68-k9	3191	3277	-2.62
B-n78-k10	3639	3579	1.68

Table 15 Compare result of PSO after applying optimization for set E

Problem	PSO	PSO with optimize	%Dev
E-n22-k4	618	606	1,98
E-n23-k3	1003	1017	-1,38
E-n30-k3	1093	1191	-8,23
E-n33-k4	1346	1332	1,05
E-n51-k5	1384	1424	-2,81
E-n76-k7	2214	2223	-0,4
E-n76-k8	2202	2223	-0,94
E-n76-k10	2244	2268	-1,06
E-n76-k14	2340	2369	-1,22
E-n101-k8	2993	3048	-1,8
E-n101-k14	3184	3171	0,41

Table 16 Compare result of PSO after applying optimization for set F

Problem	PSO	PSO with optimize	%Dev
F-n45-k4	1893	1948	-2,82
F-n72-k4	1091	1088	0,28
F-n135-k7	5794	5817	-0,4

Table 17 Compare result of PSO after applying optimization for set M

Problem	PSO	PSO with optimize	%Dev
M-n101-k10	3510	3560	-1,4
M-n121-k7	5806	5779	0,47

M-n151-k12	4598	4630	-0,69
M-n200-k16	6119	6149	-0,49
M-n200-k17	6143	6138	0,08

Table 18 Compare result of PSO after applying optimization for set P

Problem	PSO	PSO with optimize	%Dev
P-n16-k8	466	481	-3,12
P-n19-k2	317	340	-6,76
P-n20-k2	336	352	-4,55
P-n21-k2	362	374	-3,21
P-n22-k2	376	365	3,01
P-n22-k8	736	724	1,66
P-n23-k8	639	634	0,79
P-n40-k5	1004	1057	-5,01
P-n45-k5	1245	1265	-1,58
P-n50-k7	1312	1328	-1,2
P-n50-k8	1370	1382	-0,87
P-n50-k10	1365	1417	-3,67
P-n51-k10	1455	1527	-4,72
P-n55-k7	1453	1470	-1,16
P-n55-k10	1523	1545	-1,42
P-n55-k15	1633	1638	-0,31
P-n60-k10	1717	1734	-0,98
P-n60-k15	1825	1822	0,16
P-n65-k10	1945	1927	0,93
P-n70-k10	2070	2112	-1,99
P-n76-k4	2152	2179	-1,24
P-n76-k5	2157	2173	-0,74
P-n101-k4	2982	3007	-0,83

VI. Conclusion

In this report, a revised PSO algorithm has been presented for solving CVRP. We have implemented the algorithm and run it through the six sets of data. So, the result given received are not good. In the future work could be improvements such as applying some TOE, TOI and TSPOE used to improve solutions or finding some approach in a way that performs well on large benchmarks too.

References

[1] https://en.wikipedia.org/wiki/Vehicle_routing_problem

[2] https://how-to.aimms.com/Articles/332/332-Formulation-CVRP.html

[3] https://en.wikipedia.org/wiki/Particle_swarm_optimization

[4] Augerat, P., Belenguer, J.M., Benavent, E., Corbern, A., Naddef, D., & Rinaldi, G., “Computational results with a branch-and-cut code for the capacitated vehicle routing problem,” Res. Rep. 949-M, Univ. Joseph Fourier, Grenoble, Fr., 1995.

[5] Christofides, N. and S. Eilon, “An Algorithm for the Vehicle-dispatching Problem,” J. Oper. Res. Soc., vol. 20, no. 3, pp. 309–318, Sep. 1969.
[6] Fisher, M. L., “Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees,” Oper. Res., vol. 42, no. 4, pp. 626–642, 1994
[7] Christofides, N., Mingozzi, A., & Toth, P., “The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, & C. Sandi (Eds.), Combinatorial optimization. Chichester: Wiley.,” pp. 315–338, 1979.
[8] M. Clerc; J. Kennedy, “The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space,” pp. 58-73, 2022.