Received: July 3, 2020. Revised: July 29, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Solving Capacitated Vehicle Routing Problem Using Chicken Swarm Optimization with Genetic Algorithm

Nora Niazy[1,2]* **Ahmed El-Sawy[1]** **Mahmoud Gadallah[2]**

[1]*Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Egypt*
[2]*Modern Academy for Computer Science and Management Technology, Egypt*
* Corresponding author's Email: nora_niazy@fci.bu.edu.eg

Abstract: The Capacitated Vehicle Routing Problem is the most popular type of Vehicle Routing Problem and is a kind of NP-hard problems. Finding the minimum total distance travelled by vehicles to serve a group of customers with respect to capacity constrains is the aim of the Capacitated Vehicle Routing Problem. This problem will be solved by hybrid algorithm combining Chicken Swarm Optimization algorithm with Genetic Algorithm using Crossover and Mutation operation. The main idea of the proposed algorithm is to use the hieratical order of Chicken Swarm Algorithm to find paths after using the moving equations. Then we will rearrange the hieratical order according the paths cost. In an attempt to improve results for some chickens, we will use the Genetic Algorithm because it has the advantage that it searches in the neighbourhood to find the best solution then we will get the best solution which has the lowest cost. Results from a computational experiment on 10 different datasets show that the hybrid algorithm can be considered as an efficient approach and overcome the best known results in 10 datasets which means that it is 100% better than best known results which exist on NEO benchmark.

Keywords: Capacitated vehicle routing problem, Chicken swarm optimization, Genetic algorithm, Particle swarm optimization, Mutation, Crossover.

1. Introduction

One of the most important challenges facing people today is the problem of transportation. Many organizations spend a lot of money trying to reduce the cost resulting from using vehicles to deliver goods to their customers. The Vehicle Routing Problem (VRP) can be considered as one of the popular kind of transportation problems. Although the vehicle routing problem has been studied for more than 70 years, the challenge of solving it has increased as more difficult variants arise in the prominent area of development [1]. The VRP is used to find and discover the shortest path for vehicles that serve a group of customers according to some conditions that must be taken into account while performing the service. This type of problem is considered as difficult and very complex problem (NP-hard Problem) that has attracted many scientists and researchers to find multiple ways to solve it [1, 2]. In Fig. 1, we can see a simplified form of the VRP in which a path is drawn for a vehicle or a group of vehicles, this path is characterized as the shortest path that can be taken with the service of all customers, noting that the path starts and ends at the same distribution point. The problems of the VRP divided into many forms according to the restrictions placed on the problem. These restrictions may relate to time, customers, type of goods transported, etc.

One of the most important types of VRP is Capacitated VRP (CVRP), in this type, pathways to customer service are specified and each path contains a group of customers who are serviced by a specific vehicle. It must be taken into account that each customer service is done once through the specific vehicle. Also, it must be taken into account that the customer requirements do not exceed the storage space of the vehicle. The second type that we have is VRP with Time Window (VRPTW), in this type we must consider the timing assigned to each customer.

Received: July 3, 2020. Revised: July 29, 2020.

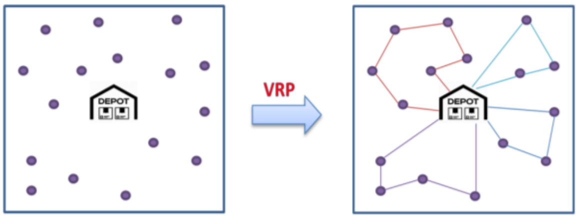


Figure. 1 The VRP is finding a shortest path for vehicles to serve some customers

One of the famous types is VRP with Pickup and Delivery (VRPPD), this type is characterized by accepting the possibility for the customer to return the goods. With this type, the capacity of each vehicle must be able to accommodate the requirements and returns of customers. The last type we will mention is Distance Constraint VRP (DCVRP), this type defines a specific distance for each vehicle so that the specified paths do not exceed the maximum specified distance for the vehicles [3]. In this paper we will solve the CVRP by using hybrid algorithm between Chicken Swarm Optimization (CSO), Tabu Search (TS) and Genetic Algorithm (GA) which give us the advantage of CSO in the chickens' various movements can be led to achieve a good balance between randomness and determinism for finding the optimum, the second advantages are that the whole chicken swarm consists of several groups, namely multi-swarm. Through integration of the hierarchal order, chickens of the different groups may behave as a team and coordinate themselves to forage for food. Thus CSO can behave intelligently to optimize problems efficiently [4]. Also we use the power of TS to avoided local optima where all the neighboring solutions are non-improving [5], also we will use the advantages of GA in applying an iterative and stochastic process on a group of individuals (population), where each individual represents a potential solution to the problem [3]. By using previous advantages for CSO, TS and GA we will get good results that surpass the current best known results. In this work we will propose a hybrid algorithm CSO with Tabu Search and Genetic Algorithm (Crossover and Mutation operations both or one of them). We will mention the basic constraints for CVRP in next section then in section 3 we will talk about the history of the problem as well as the algorithms that we will use to solve the problem.in section 4 we describe how we apply the proposed hybrid algorithm to solve the problem. The experimental results and the comparing between our hybrid algorithm with CSO, Particle Swarm Optimization PSO, and best known results and which algorithm is better will be mentioned is sections 5.

2. Problem formulation

We can consider the problem of VRP is a combinatorial problem and it can be represented as complete directed graph $G(V,E)$, where $V=\{V_d, V_c\}$ is a group of vertices including the clients(V_c) and the depots or warehouses (V_d) and E for the connection lines between the customers [6]. Despite the many types of VRP, the problem of CVRP remains the most widespread and studied type of VRP. However, despite the large number of studies that were studied on CVRP, it remains a computationally complex problem and the greater the scale of the problem, the greater the time required to solve it greatly, knowing that the accuracy of the solutions depends on the inputs of the problem in addition to the time given to solve it. We can consider the number of customers, only one central depot, the number of vehicles, the same capacity of these vehicles and the same product are the main components for CVRP. The goal of the CVRP is to put a set of paths that serve all customers through a group of vehicles, and when finding these paths should be as short as possible and also the cost of these paths should be as least as possible, taking into account that each vehicle serves a specific group of customers so that each customer is served once by one vehicle, starting and ending at the same warehouse and the last important thing, the total requirements of customers in each group should not override the storage space for each vehicle [7]. The CVRP can be considered as an undirected graph $G=(V,E)$ where $V = \{v_0, v_1, . . . , v_n \}$ is a vertex group and $E=\{(v_i, v_j)/v_i, v_j \in V, i < j\}$ is an edge group. We can consider the depot as Vertex v_0 , and it is from where y correspondent vehicles of capacity should be able to serve every customer, dealing with a group of n vertices $\{v_1, . . . , v_n\}$. We determine on E (a non-negative cost) distance matrix $X=(X_{ij})$ between customers v_i and v_j . Let $V_1,..,V_m$ be a split of V, a path P_i is a permutation of the customers in V_i assigning the order of visiting them, starting and finishing at the depot v_0. The cost of a given path $P_i=\{v_{i0}, v_{i1}, ... , v_{ik+1}\}$, where $VIJ \in V$ and $v_{i0} = v_{ik+1} = 0$ (0 indicates the depot, i indicates the number of customer)[3,8]. Is given by:

$$Cost(Pi) = \sum_{j=0}^{k} x_{j,j+1} \tag{1}$$

And the problem solution F(S) is:

$$F_{CVRP}(S) = \sum_{i=1}^{y} Cost(Pi) \tag{2}$$

The CVRP consists of defining a group of y vehicle paths:

The smallest total cost.

Starting and stopping at the depot v_0.

Every customer is visited just once by a specific vehicle; according to the restrictions.

Any path doesn't exceed the total requirements:

$$Q\ (\sum_{v_j \epsilon P_i} q_{j<Q}) \qquad (3)$$

The total distance of any path is not larger than a preset specific

$$T(Cost(Pi)\ \leq T) \qquad (4)$$

For all clients the type of product should be the same [3].

The number of vehicles is one of the important elements that effect on the decision taking and the cost of problem. In this work, the path length is reduced separately of the number of used vehicles[9].

3. Literature review

We can consider the problem of the VRP a very complex problem and this requires a great time to solve it, and over time a lot of theories have been developed to solve this problem and we will list a set of these solutions that some researchers have come up with. Various methods have been introduced to solve VRP, they can be categorized as exact methods, heuristic algorithms, and metaheuristic algorithms. Exact methods can solve small and medium VRP instances. According to this Constraint, almost algorithms used to solve VRP are heuristic and metaheuristic. Both heuristic and metaheuristic algorithms propose approximate solutions in reasonable computing times, so they are more practical for real-world cases and commercial applications[1]. The next algorithms that we will mention depend on heuristic techniques. a local version PSO based on neighbourhood operator proposed by Suganthan (1999), in the particle swarm optimization algorithm, each particle timely adjusts the position in the searching space according to its flying experience and neighbour node's flying experience [10]. For solving CVRPs Baker. suggest a simple Genetic Algorithm (GA). Bell. introduce Ant Colony Optimization (ACO) algorithm for solving the CVRP. For solving great problems (more than 100 clients) the authors suggest a multiple-ant-colony strategy. An approach which combined two algorithms ACO and Scatter Search was introduced by Zhang. for solving CVRP. In 1995 Eberhart and Kennedy propose a PSO. Their approach is depending on two-stage technique. The first stage is using DPSO to apply the task of clients clustering and the second stage is applying Simulated Annealing (SA) to determine the visiting order for every vehicle [11]. In the particle swarm optimization algorithm, each particle timely adjusts the position in the searching space according to its flying experience and neighbour node's flying experience. The particle swarm is randomly initialized, and the algorithm aims at searching the optimum solution which meets some performance [10]. Robinson, Sinton, and Rahmat-Samii (2002) investigated the possibility of hybridizing PSO and GA to optimize the design of a profiled corrugated horn antenna. They proposed two approaches - GA-PSO and PSO-GA. In GA-PSO approach, GA is applied till improvement in objective function evaluation started to level off and then the GA output is used as the input to PSO [12]. Literature (Xu, Lu, & Cheng, 2017) Suggested that a sub-path be created and this sub-path contributes to creating the best path and then moving to the second phase of the pheromone application, and use random interpolation method to set the ranking of cities in the optimal solution. It improves the convergence speed and obviate the algorithm falling into local optimal [13]. In the previous works and researches, a single depot with customers distributed around this depot is supposed. The route that was specific for a vehicle, start and end in a centre depot. But sometimes there may be more than one depot and in this case it must be handled by planning various routes covering all the clients or nodes[7, 14]. The majority of the techniques used are heuristic techniques, but the proposed algorithm is a Meta-heuristic algorithm in addition to it is Multi-Swarm Algorithm, which is mean that each sub-swarm focus on specific region while a specific diversification method and this feature gives us a preference in the speed and accuracy of solutions. In a previous work [9], we proposed a hybrid algorithm (CSO with TS) for CVRP, in which more details were mentioned in terms of solving each algorithm for the problem separately, but here we proposed a hybrid CSO,TS and GA in an attempt to obtain much better results than the previous. We will compare the implementation of PSO, CSO and the hybrid CSO with GA algorithms on NEO benchmark datasets. It will become evident to us through the results that the proposed algorithm has exceeded the results of the PSO.

3.1 Chicken swarm optimization (CSO)

The CSO was proposed in 2014 by Meng and this algorithm can be considered as Optimization algorithm, this algorithm is inspired by the natural life of chickens. The manner of the chicken swarm reckon on hierarchal order. The chicken swarm can be divided into many groups, each group contains one rooster and many hens and chicks. There exist competition between various chickens under certain hierarchical order [4]. Dependence on the hierarchical formation within the swarm is such that this formation is topped by the highest fitness values and in this case this formation is topped by the roosters, and those with the worst fitness values are at the end of the formation and in this case we can consider them the chicks. In the same time, those in the middle are hens. The swarm is divided into groups, each group containing a rooster, a group of hens and a group of chicks and they are generated randomly. The rooster with the highest fitness value can search for food in more places and on a larger scale [15, 16] .

$$x_{i,j}^{t+1} = x_{i,j}^{t} \times (1 + Randn(0, \sigma^2)) \quad (5)$$

where $x_{i,j}^{t+1}$and $x_{i,j}^{t}$ are the position of jth dimension of particle i in t+1 and t iterations, respectively, and randn(0, σ) is a random number of Gaussian distribution whose variance is σ^2.The parameter σ^2 can be calculated [16].

$$\sigma^2 = \begin{cases} 1, \text{if } f_i \le f_j \\ e^{\left(\frac{(f_k - f_i)}{|f_i| + \varepsilon}\right)}, otherwise\ k \in [1, N], k \ne i \end{cases} \quad (6)$$

Where i, j ∈ [1, $r$size] and $i \neq$ j. $r$size represents the number of rooster swarms. f_i and f_j are the fitness values of rooster i and j, respectively; ξ represents a number which is few adequate [16].
Some hens can rob good food from another group

$$x_{i,j}^{t+1} = x_{i,j}^{t} + S1 \times Rand \times \left(x_{r1,j}^{t} - x_{i,j}^{t}\right) + S2 \times Rand \times \left(x_{r2,j}^{t} - x_{i,j}^{t}\right) \quad (7)$$

Where $x_{r1,j}^{t}$ and $x_{r2,j}^{t}$ are the position of rooster individual $r1$ in the population of hen xi and rooster individual $r2$ in the other population, respectively. Rand is a uniform random number over [0, 1]. *S1* and *S2* indicate the weight calculated [16].

$$S1 = e^{\left(\frac{(f_i - f_{r1})}{|f_i| + \varepsilon}\right)} \quad (8)$$

And

$$S2 = e^{(f_{r2} - f_i)} \quad (9)$$

Where f_{r1} and f_{r2} are respectively, the fitness value of rooster individual $r1$ in the population of hen xi and rooster individual $r2$ in the other population [16]. Chicks search for food beside their mothers

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL \times \left(x_{m,j}^{t} - x_{i,j}^{t}\right) \quad (10)$$

Where FL stands for a parameter, meaning that the chick would follow its mother to forage for food. $x_{m,j}^{t}$ represents the position of the i th chick's mother (m ∈ [1,N]) [16].

3.2 Genetic algorithm (GA)

The Genetic Algorithm (GA) technique is a group of computational processes model stimulated by evolution. This algorithm uses a chromosome-like data construction to find a potential and optimal solution for a given specific problem. The GA was first defined and developed by J. Hollan, also the GA can be considered as an evolutionary algorithm appropriate for solving such scheduling problems [3, 17] and it is a very good optimization technique that emphasizes through several phase like selection, crossover and mutation to detect an optimal solution. Genetic Algorithm simulates the procedure of natural selection using bio-inspired operators such as crossover and mutation.

It depends on the theory of survival of the fittest. Genetic Algorithm considers a population of solutions (individuals). The fitness of each solution is determined by evaluating a fitness function against each solution. The survival of the individual to the next iteration is absolutely based on the fitness value of the individual. The individuals with least fitness value will be rejected from the population. Great feature gain from parent solutions are propagated to the next generation by applying crossover and mutation [7]. The pseudo code for evolutionary algorithm is shown in the Fig. 3.

Crossover is a genetic operator used to turn the chromosomes from one generation to the next. It is definition to reproduction and biological crossover, onto which genetic algorithms are based. Both implemented crossovers don't do mutual exchange of genetic material between two parents. The creation of new child depends on the information that was taken from one individual and inserts it in the other individual. Also in genetic algorithms, mutation is a genetic operator used to preserve genetic difference from one generation of a population of chromosomes to the next. It is identical to biological mutation. The probability which mutations will take place and if mutation takes place at all can be configured [3].

Received: July 3, 2020. Revised: July 29, 2020.

- Initialize a population
- Fitness function evaluation
- Do until Max_Gen
  - For each G group
  - Order chicken based fitness value
  - Determine of rooster and hens and chicks
  - For J=1 :N
    - Update rooster if J=rooster
    - Update hen if J=hen
    - Update chick if J=chick
    - Calculate fitness function for new population
    - IF new population is better ,update previous one

Figure. 2 Chicken swarm optimization pseudo code

3.3 Particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) is a global optimization technique proposed by Kennedy and Eberhart (1995). A swarm consist of a group of particles that each particle represents a possible solution.

Suppose that each solution can be considered as a point in N-Dimensional space that each point or particle has start velocity, particles move through solution space, and after each time step, particles are rated according to some fitness criterion. They are accelerated towards particle with best fitness value within their communication group. These movies of PSO help particles escape from local optimal solutions. Each particle has a simple memory that remember the position of best solution achieve by itself, this value is called personal best (*pbest*) and the position of best solution obtained so far by any particle in the neighbourhood 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 [18].The formulas for updating the velocity of each particle in the swarm are as follows.

$$\boldsymbol{V_{id}\ (t+1) = W\ V_{id}\ (t) + c1r1(P_{id} - X_{id}) + c2r2(P_{gd} - X_{id})} \qquad (11)$$

Where

V$_{id}$: velocity of dimension d of the i th particle.
P$_{id}$: personal best previous position of the i th Particle.
P$_{gd}$: the global best position for all particles.
X$_{id}$: current position of the i th particle.
c1 & c2 : are acceleration constants.
r1, r2 : random function in the range [0, 1].
W : Inertia weight.

1. *P* ← Generate Initial Population ();
2. Evaluate (*P*);
3. **while** !Stop Condition() **do**
4. *P´* ← Select Parents (*P*);
5. *P´* ← Apply Variation Operators *(P´);*
6. Evaluate *(P´);*
7. *P* ← Select New Population *(P,P´);*
8. **end while**
9. **Result:** The best solution found

Figure. 3 The pseudo code for genetic algorithm

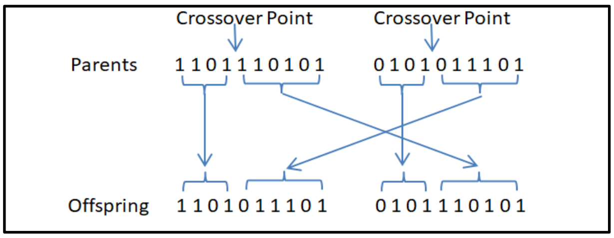


Figure. 4 Crossover example

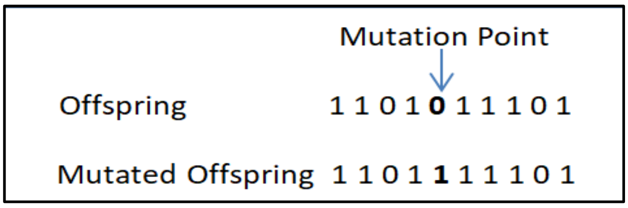


Figure. 5 Mutation example

The new velocity calculated in that iteration is used for updating the current position of each particle by using the following formula:

$$X_{id}\ (t+1) = X_{id}\ (t) + V_{id}\ (t+1) \qquad (12)$$

Where

X$_{id}$ (t) : current position of the i th particle.
X$_{id}$ (t+1) : new position of the i th particle.
V$_{id}$ (t+1) : New velocity of the i th particle.

The process of updating the velocity and the position of each particle in the swarm is repeated until it reached a specific number of iterations as a termination criteria or reaching to the same final solution [18].

4. Proposed work

CVRP will be solved by using new hybrid Algorithm CSO, Tabu Search and Genetic Algorithm (Crossover and Mutation operations both or one of them) (HYCSOGATS) which will be listed in the next sections.

 DOI: 10.22266/ijies2020.1031.44

Received: July 3, 2020. Revised: July 29, 2020.

4.1 Hybrid CSO with GA and TS for CVRP

The CVRP will be solved in this part by using CSO with GA and TS, and as we mentioned earlier, one of the advantages of using TS is that it helps to avoid local optima where all the neighbouring solutions are non-improving. This advantage will be used with roosters and dominated hens to get better results. Also we will use the advantages of GA in applying an iterative and stochastic process on a group of individuals (population), where each individual represents a potential solution to the problem [3]. To measure their aptitude for the problem, the individuals are assigned a fitness value. This value represents the quantitative information used by the algorithm to guide the search. The trade-off between exploration of new areas of the search space and exploitation of good solutions accomplished by this kind of algorithms is one of the key factors for their high performance with respect to other meta-heuristics. This exploration/exploitation balance can be sharpened with some different parameters of the algorithm such as the population used (decentralized or not), the variation operators applied, or the probability of applying them, among others. In the first, initialize a population of chickens and define related parameters as flow (The number of iterations - the number of populations in the swarm the number of each roosters, hens, and chicks in the population – the coordinates of each customer – the capacity of vehicles – specify some constants like flow mother - the initial solutions- Crossover Percentage- Number of Offsprings- Mutation Percentage- Mutation Rate). After that we will generate a random solution and calculate the fitness for this solution, then we apply the Tabu Search and calculate the fitness, after that we choose the best fitness value. According to previous results we will update the hierarchical order for chickens. After initialization steps the next steps repeated until the number of iterations is finished. In this case we have some roosters, hens, and chicks. We will apply CSO moving equations on the current iteration. Then we apply the Crossover operation on hens only and the Mutation operation on chicks only according to the chosen operation to get t solution, which means if the chosen operation is the Crossover then the algorithm used will be (HYCSOCRTS) or if the chosen operation is the Mutation then the algorithm used will be (HYCSOMUTS) or if the chosen operation is both (Crossover & Mutation) then the algorithm used will be (HYCSOGATS). Then we calculate the fitness value for t solution and compare it with the fitness of the previous iteration solution and choose best solution which has best fitness value and so on. The algorithm is described as follows in Fig. 6.

5. Experimental results

In this section, we present the implementation of our proposed methodologies. Experimental results using different sets of parameters are shown along with explanations for the results values. A comparison between Particle Swarm Optimization (PSO), Chicken Swarm Optimization (CSO), Hybrid CSO with genetic operators (Crossover and Mutation) (HYCSOGATS), Hybrid CSO with Crossover (HYCSOCRTS) and Hybrid CSO with Mutation (HYCSOMUTS) results for solving CVRP also provided. The comparison is made upon the results of experiments applied on well-known benchmarks. Our methodology is implemented using the following technologies.

Software: Matlab R2013a, Windows 10 Pro 64-bit operating system, Microsoft Excel.

Hardware: Intel(R)Core(TM)i7-5500U@2.40GHz machine, 16 GB RAM, Intel(R)HD Graphics 5500, AMD Radeon(TM) R9 M375.

We will compare the proposed algorithms HYCSOGATS, HYCSOCRTS and HYCSOMUTS algorithms by using three different classes of NEO benchmark datasets Augerat (A, B, P) of CVRP with instances per class [8] also with CSO [9] , PSO . The instance details are shown in the Table 1.

From Table 1 it appears that in class A; 5, 6, 6, 7, and 10 vehicles are assigned to 31, 36, 44, 53, and 79 customers respectively and there is one node in each instance for warehouse and the capacity for each vehicle is 100. Moreover, in class B; 5, 5 and 9 vehicles are assigned to 34, 38, and 56 customers respectively and there is one node in each instance for depot and the capacity of each vehicle is 100. Finally, in class P; 8, and 15 vehicles are assigned to 15 and 59 customers respectively and the capacity of first vehicle is 35 and for second one is 80.

The population structure is the main difference between our proposed algorithms, so we list different parameters' values used for the proposed algorithms in Table 2. In the Table 3, we compare CSO against best known results in the swarm fields and we consider the cost factor as will be indicated also as shown in the Fig. 7 and Fig. 8. As the Table 3, Fig. 7, and Fig. 8 which explain that we solve CVRP by a new technique CSO algorithm which is solved completely and we observe that the PSO algorithm is better than CSO algorithm. We indicate to solve a minimization problem so, in our graphs less results is better for that, we tried to enhance our results of CSO by combining it with genetic operators (Crossover, At T = 0

1. Initialize the number of customers and the number of vehicles
2. Initialize the dimension of the problem , the position of each customer and the position of depot
3. Initialize the demand of each customer and the capacity of vehicle.
4. Initialize a population of N chickens randomly and define the related parameters according to the previous initialization of the problem (the No. of chickens represent the No. of customers) and Initialize randomly ***t*** solutions of (***Ni***).
5. Initialize the Number of Offsprings for Genetic Algorithm
6. Evaluate the N chickens' fitness values
7. OP = Genetic Operation selection (Crossover or Mutation or both)
8. Improve the solution using Tabu search
9. Set or update The ***N*** chicken's fitness values if it is better than the randomly one.
10. Do until (T < Max_Gen)
    - ❖ IF (T% G == 0)
        - ➢ Order the chickens' fitness values and establish a hierarchical order in the swarm.
        - ➢ Split the swarm into different groups, and determine the relationships between chicks and mother hens in a group

          End IF
    - ❖ For i=1 : N
        - I. IF i == R Update its solution/location using Eq. (5) & Eq. (6) End IF
        - II. IF i == H Update its solution/location using Eq. (7) & Eq. (8) & Eq. (9) End IF
        - III. IF i == C Update its solution/location using Eq. (10) End IF
        - IV. IF the selected operator is Mutation and i= C then
            - ♦ Initialize a Mutation Percentage - Mutation Rate
            - ♦ Apply Mutation

          ELSE

          IF the selected operator is Crossover and i= H then
            - ➢ Initialize a Crossover Percentage
            - ➢ Apply Crossover

          ELSE

          IF the selected operator is both
            - ▪ Initialize a Mutation Percentage - Mutation Rate- Crossover Percentage
            - ▪ IF i== R then go to step (V)
            - ▪ IF i== H then perform crossover
            - ▪ IF i== C then perform mutation
        - V. Calculate the fitness value
        - VI. Set or update The ***N*** chicken's fitness values if it is better than the previous one.
        - VII. Evaluate the new solutions by calculating the summation of each vehicle cost (S').
        - VIII. If the new solution (S') is better than its previous one (S) update it (S = S')

          End for
    - ❖ T = T+1
    - ❖ Go to step 10
11. Get S

Figure. 6 HYCSOGATS for solving CVRP

Mutation) and TS as a hybrid algorithm which appears also in the Table 3. As the Table 3 and Fig. 7 and Fig. 8 we found that the costs of hybrid CSO with mutation, TS algorithm and hybrid CSO with Crossover, Mutation, TS algorithm are always better than PSO. We compare our obtained results against to the best know results in the literature as any accurate work should do. Also as shown in Table 3 and Fig. 8 the best cost of Hybrid CSO with Mutation, TS algorithm and Hybrid CSO with Crossover, Mutation, TS Algorithm is always better than benchmark cost [8] and it's overcome the benchmark results through 100 runs represented as Hit rate and our results has overcome the best results known (benchmark) completely. For CSO and hybrid CSO with Crossover, TS algorithm the costs also better than benchmark results except 2 instances C7 and C10. Also in Fig. 9 which contain a comparison between all algorithms in this paper by using relation between average cost for 100 run and number of iterations (1000 Iterations), we can observe that the HYCSOMUTS and HYCSOGATS is always better than the PSO.

Received: July 3, 2020. Revised: July 29, 2020.

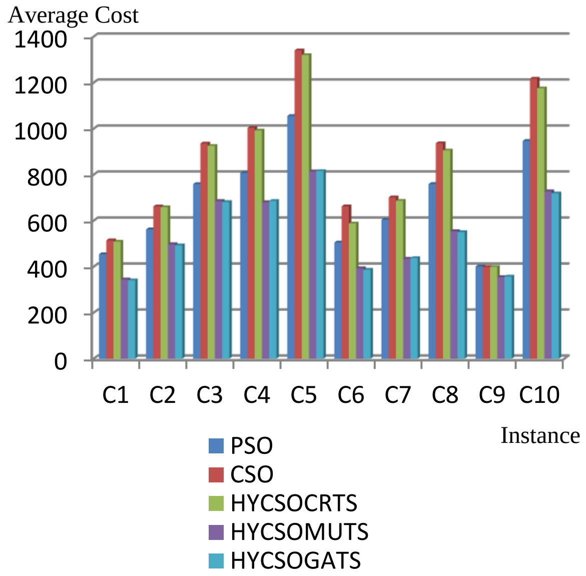


Figure. 7 Average costs for PSO, CSO, and hybrid CSO algorithms (less is better)

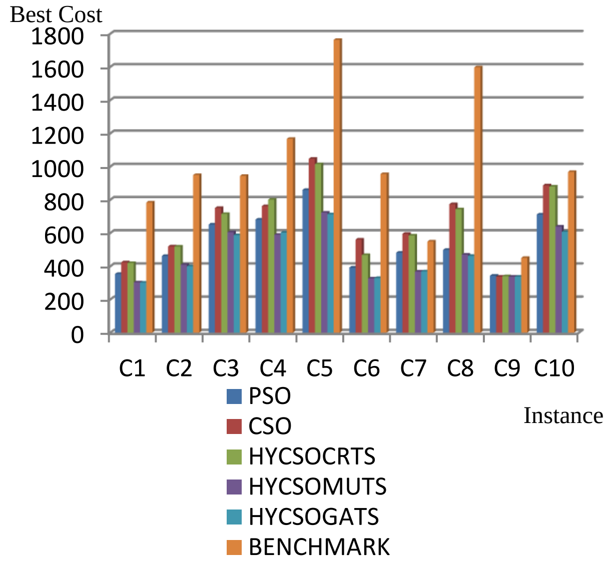


Figure. 8 Best costs for benchmark, PSO, CSO, and hybrid CSO algorithms (less is better)

Table 1. The instances details

Instance Symbol	Problem Name	No. of Nodes	No. of Customers	Vehicles count	Vehicle capacity	Type
C1	A_N32_K5	32	31	5	100	A
C2	A_N37_K6	37	36	6	100	A
C3	A_N45_K6	45	44	6	100	A
C4	A_N54_K7	54	53	7	100	A
C5	A_N80_K10	80	79	10	100	A
C6	B_N35_K5	35	34	5	100	B
C7	B_N39_K5	39	38	5	100	B
C8	B_N57_K9	57	56	9	100	B
C9	P_N16_K8	16	15	8	35	P
C10	P_N60_K15	60	59	15	80	P

Table 2. Parameters' values

Parameter	**Value**
Number of iterations	1000
Population size	100
Dimensions	From instances
PSO	
Inertia Weight	1
Inertia Weight Damping Ratio	0.99
Personal Learning Coefficient	2
Global Learning Coefficient	2
Maximum Velocity	0.1
Minimum Velocity	-0.1
CSO	
Roosters Percentage	15 %
Hens Percentage	70 %
Chicks percentage	15%
Genetic	
Crossover Percentage	0.7
Extra Range Factor for Crossover	0.4
Mutation Percentage	0.3
Mutation Rate	0.01

6. Conclusion

In this paper we provide a new solution to the problem of capitated vehicle routing problem (CVRP) by using a hybrid algorithm that integrate between CSO algorithm and the GA. CVRP is an

Received: July 3, 2020. Revised: July 29, 2020.

Table 3. Comparison between PSO results and hybrid CSO

Instance Symbol	Algorithm	Mean Cost	Maximum Cost	Minimum Cost	Standard Deviation	Hit rate %	Best Known (Benchmark)
C1	PSO	454.307033	559.7693	353.534	46.00261	100	784
	CSO	514.604359	594.6326	423.6803	37.65053	100	
	HYCSOCRTS	509.1024	622.4166	419.3538	41.73381	100	
	HYCSOMUTS	344.8993	432.1224	302.7995	26.23513	100	
	HYCSOGATS	340.7837	400.5959	301.0805	19.46375	100	
C2	PSO	562.4223	658.0035	461.8879	37.40148	100	949
	CSO	662.0233	845.8991	519.5303	57.8781	100	
	HYCSOCRTS	658.8917	847.1445	518.3605	55.32069	100	
	HYCSOMUTS	498.089	632.4186	408.8499	36.44005	100	
	HYCSOGATS	492.9874	607.6489	399.3542	45.33578	100	
C3	PSO	759.4891	905.6364	651.3552	58.32224	100	944
	CSO	935.632	1306.743	750.6594	92.86331	56	
	HYCSOCRTS	925.3248	1236.854	714.7642	86.53806	62	
	HYCSOMUTS	685.710515	766.3425	605.1577	34.20851	100	
	HYCSOGATS	681.274559	799.9469	587.4775	41.48588	100	
C4	PSO	807.6556	946.2357	681.1021	59.03508	100	1167
	CSO	1003.5489	1314.438	761.9232	98.07118	94	
	HYCSOCRTS	992.333163	1248.799	800.8694	88.44271	97	
	HYCSOMUTS	680.6451	789.9855	589.5107	34.73992	100	
	HYCSOGATS	685.8487	817.5731	602.0497	34.00751	100	
C5	PSO	1055.457	1254.06	859.3825	83.94576	100	1763
	CSO	1340.357	1826.731	1046.661	153.8499	98	
	HYCSOCRTS	1320.156	1774.346	1014.775	136.1512	99	
	HYCSOMUTS	814.58946	902.2708	722.6783	42.59461	100	
	HYCSOGATS	815.543415	1005.318	712.7805	45.91423	100	
C6	PSO	505.217	620.899	391.6075	37.74062	100	955
	CSO	662.569199	758.0507	560.7559	42.98174	100	
	HYCSOCRTS	587.88933	699.4382	467.1473	48.18598	100	
	HYCSOMUTS	393.6137	467.3745	326.364	31.93803	100	
	HYCSOGATS	386.6958	495.8733	328.8926	32.79038	100	
C7	PSO	604.399591	705.7596	481.1474	46.84487	10	549
	CSO	701.655	793.5782	594.9109	44.74526	0	
	HYCSOCRTS	686.3594	789.4614	584.7762	44.70158	0	
	HYCSOMUTS	434.608005	521.3993	367.8033	36.78653	100	
	HYCSOGATS	437.528519	553.2995	368.995	35.39143	99	
C8	PSO	759.5568	950.7142	498.2958	61.82116	100	1598
	CSO	936.621093	1179.3586	773.5207	77.49575	100	
	HYCSOCRTS	905.559	1113.983	743.1081	67.605	100	
	HYCSOMUTS	554.781161	652.1262	469.644	42.96788	100	
	HYCSOGATS	550.759535	644.9764	462.3205	40.63339	100	
C9	PSO	401.074	555.9745	342.4674	40.65797	89	450
	CSO	398.96339	483.3567	336.925	37.76741	88	
	HYCSOCRTS	399.026387	497.7501	339.6567	36.49407	88	
	HYCSOMUTS	355.08111	426.5276	336.925	16.30825	100	
	HYCSOGATS	357.222	439.1826	336.925	20.99526	100	
C10	PSO	947.0628	1408.375	711.2227	110.496	59	968
	CSO	1217.715	1613.475	886.7962	165.5583	5	
	HYCSOCRTS	1175.446	1494.49	879.4456	143.83	9	
	HYCSOMUTS	727.8232	845.5919	638.7676	46.1496	100	
	HYCSOGATS	718.892588	811.3717	606.8586	43.02001	100	

Received: July 3, 2020. Revised: July 29, 2020.

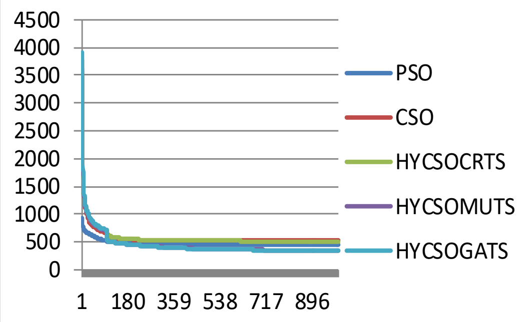
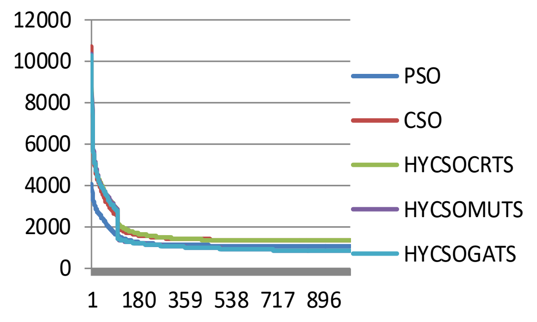
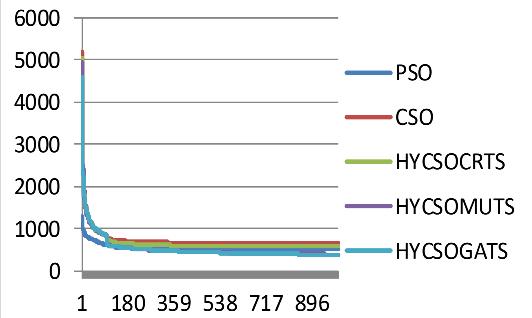
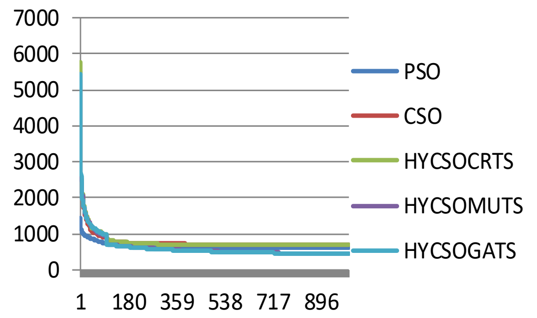
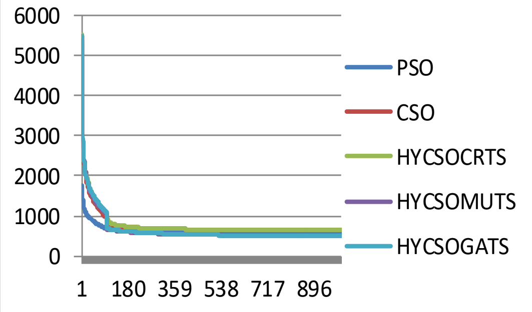
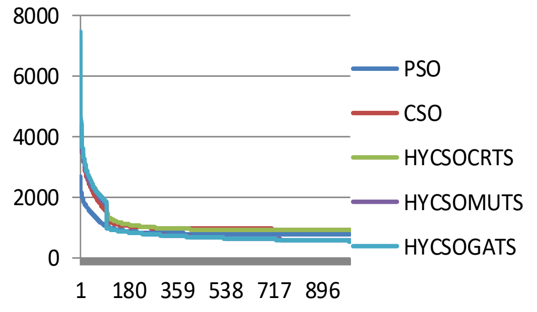
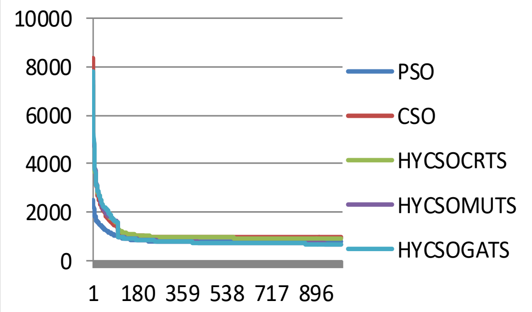
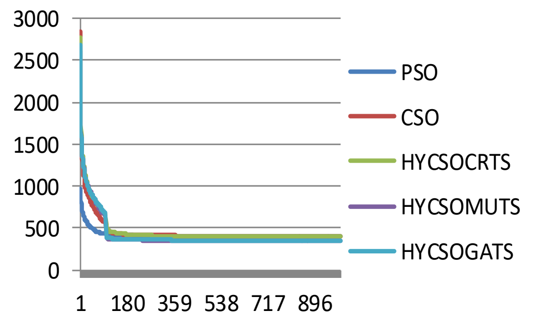
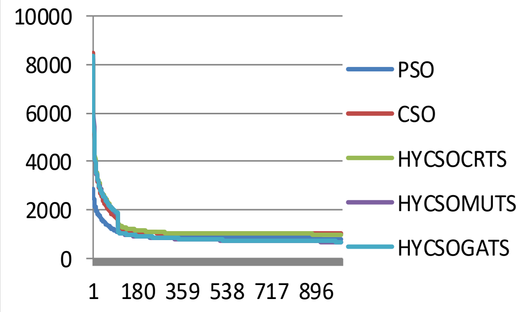
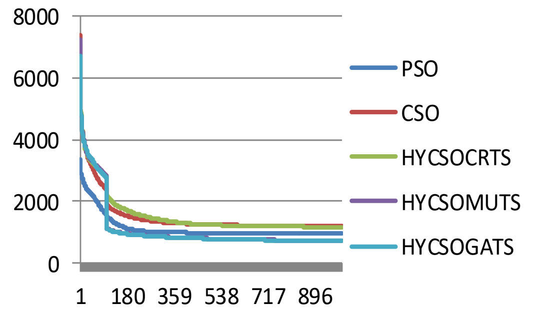
Instance	Graph	Instance	Graph
C1		C6	
C2		C7	
C3		C8	
C4		C9	
C5		C10	

Figure. 9 Between algorithms by using relation between average cost and number of iteration

integer programming problem, which is fall into the category of NP- hard problem with the goal of minimizing the total travelled distance by the involved capacity – limited vehicles to reach their destination and back to the start depot. We also compared the results CSO against PSO in solving CVRP, from the results we found that the PSO algorithm is better in cost than CSO algorithm. So we improved our solution by create a new hybrid algorithm by combining CSO with TS and GA using a Crossover and Mutation operators (HYCSOGATS). Then we found the best cost of (HYCSOGATS) is always better than benchmark and PSO costs by 100%.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Nora, Ahmed, and Mahmoud; methodology, Ahmed; software, Nora; validation, Nora, Ahmed, and Mahmoud; formal analysis, Ahmed, and Mahmoud; investigation, Nora; resources, Nora, and Ahmed; data curation, Nora; writing—original draft preparation, Nora, and Ahmed; writing—review and editing, Nora, and Mahmoud; visualization, Nora, and Ahmed; supervision, Mahmoud.

References

[1] R. Linfati and J. Escobar, "Reoptimization Heuristic for the Capacitated Vehicle Routing Problem" , *Journal of Advanced Transportation*, Vol. 2018, No. 3743710, pp. 1–8, 2018.

[2] S. Kumar and R. Panneerselvam, "A Survey on the Vehicle Routing Problem and Its Variants", *Intelligent Information Management*, Vol. 04, No. 03, pp. 66–74, 2012.

[3] E. Alba and B. Dorronsoro, *Cellular Genetic Algorithms*, Vol.42, Springer, 2008.

[4] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A New Bio-inspired Algorithm: Chicken Swarm Optimization", In: *Proc. International Conf. in Swarm Intelligence*, Hefei, China , pp. 86–94, 2014.

[5] F. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*, Vol. 57, Springer, 2003.

[6] L. Calvet, D. Wang, A. Juan, and L. Bové, "Solving the multidepot vehicle routing problem with limited depot capacity and stochastic demands", *Int. Trans. Oper. Res.*, Vol. 26, No. 2, pp. 458–484, 2019.

[7] G. Nagy and S. Salhi, "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", *European Journal of Operational Research*, Vol. 162, No. 1, pp.128, 2005.

[8] http://neo.lcc.uma.es/vrp/vehicle-routing-problem/, 07-February-2020.

[9] N. Niazy, A. El-sawy, and M. Gadallah, "A Hybrid Chicken Swarm Optimization with Tabu Search Algorithm for Solving Capacitated Vehicle Routing Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 237-247, 2020.

[10] W. Geng, "Cognitive Deep Neural Networks prediction method for software fault tendency module based on Bound Particle Swarm Optimization", *Cognitive Systems Research*, Vol. 52, No. 2018, pp. 12–20, 2018.

[11] Y. Kao and M. Chen, "A hybrid PSO algorithm for the CVRP problem", In: *Proc. International Conf. in Evolutionary Computation Theory and Applications*, pp. 539–543, 2011.

[12] K. Bhattacharjee and M. Pant, "Hybrid particle swarm optimization-genetic algorithm trained multi-layer perceptron for classification of human glioma from molecular brain neoplasia data", *Cognitive Systems Research*, Vol. 58, No. 2019, pp. 173–194, 2019.

[13] J. Li, Y. Xia, B. Li, and Z. Zeng, "A pseudo-dynamic search ant colony optimization algorithm with improved negative feedback mechanism", *Cognitive Systems Research*, Vol. 62, No. 2020, pp. 1–9, 2020.

[14] Y. Zhang, M. Qi, L. Miao, and G. Wu, "A generalized multi-depot vehicle routing problem with replenishment based on LocalSolver", *International Journal of Industrial Engineering Computations*, Vol. 6, No.1, pp. 81-98, 2015.

[15] F. Chebihi, M. E. Riffi, A. Agharghor, S. C. B. Semlali, and A. Haily, "Improved Chicken Swarm Optimization Algorithm to Solve the Travelling Salesman Problem", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 12, No. 3, pp. 1054-1062, 2018.

[16] C. Qu, S. Zhao, Y. Fu, and W. He, "Chicken Swarm Optimization Based on Elite Opposition-Based Learning", *Mathematical Problems in Engineering Systems*, Vol. 2017, No. 6, pp. 2, 2017.

[17] A. Hussein, N. ArunKumar, G. Ramirez-Gonzalez, E. Abdulhay, J. Tavares, and V. Albuquerque, "A medical records managing and securing blockchain based system supported by a Genetic Algorithm and Discrete Wavelet

Transform", *Cognitive Systems Research*, Vol. 52, No. 2018, pp. 1–11, 2018.

[18] M. Tavakoli and A. Sami, "Particle Swarm Optimization in Solving Capacitated Vehicle Routing Problem", *Bull. Electr. Eng. Informatics*, Vol. 2, No. 4, 2013.