



A Hybrid Chicken Swarm Optimization with Tabu Search Algorithm for Solving Capacitated Vehicle Routing Problem

Nora Niazy^{1,2*} Ahmed El-Sawy¹ Mahmoud Gadallah²

¹Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Egypt

²Modern 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 optimization problems. The purpose of this problem is to decrease the total distance travelled by vehicles with respect to restrictions of vehicles' capacity. In this paper the capacitated vehicle routing problem solved by Chicken Swarm Optimization algorithm, Tabu Search and a new hybrid algorithm. The main idea of the proposed algorithm is to use the hieratical order of Chicken Swarm Algorithm and Tabu Search for finding best neighbourhood to find shortest path with minimum cost, after that we using the moving equations of the two algorithms on each chicken to construct the paths then we choose the shortest path which has the minimum 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 9 datasets which means that it is 90% better than known results.

Keywords: Vehicle routing problem, Capacitated vehicle routing problem, Chicken swarm optimization, Tabu search algorithm.

1. Introduction

The transportation business has different problems of various levels of complexity. Maybe one of the most important of these problems is route development, which is generally performed based on expert knowledge. Despite the fact that the vehicle directing problem has been read for over 70 years, these issues become more difficult over time . The vehicle routing problem (VRP), which is an extension of the well-known traveling salesman problem (TSP), is considered NP-hard within the research area of combinatorial optimization problems [1]. Optimization is a process with goal to search for the best solution among all available ones of a specific problem [2]. The goal of the VRP is to find a road that is used by the vehicles to serve a group of customers distributed in specific places and meet their needs at the lowest possible cost, this road or the specific path for the vehicle starts from a specific warehouse and ends at the same warehouse

passing through the specified customers once per customer. In general, we can consider the term VRP on any type of problem whose components consist of a warehouse or a group of warehouses in addition to a set of routes and a set of vehicles that are supposed to run on these routes in addition to a group of customers required to serve them at the lowest possible cost. We can see in the Fig. 1 a simple VRP model and a possible solution for this example. There are many different types of VRP problem, we will list the most famous as follows,

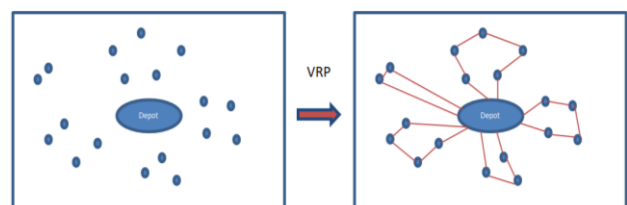


Figure. 1 The VRP is in the service of a group of clients from a depot using the routs with minimum cost [3]

- Capacitated VRP (CVRP) in this type, the vehicle's storage space must be taken into account while meeting customer requirements.
- VRP with Time Window (VRPTW) in this type, the timing of each customer must be observed.
- VRP with Pickup and Delivery (VRPPD) It should be noted in this type that the customer may return product, then the storage space inside the vehicle must be compatible with the possibility of returning the product through the customer
- Distance Constraint VRP (DCVRP) This type needs to define a certain distance for each vehicle as a maximum, and therefore this may affect the number of customers that will be served in addition to the path that these vehicles will take.

We can consider the problem of VRP as one of the problems that takes a very large arithmetic time, and the time to solve the problem increases exponentially as the size increases [3]. The problem of VRP can be solved by many theories. It is computationally complex to finding solutions for polynomial time problems and finding a globally minimum solution. As a result, to locate a near optimal solution in a reasonable amount of time for VRP we will use evolutionary computing approaches. For example: genetic algorithm, Ant Colony Optimization and Particle Swarm Optimization, Chicken Swarm Optimization (CSO) [4,5]. In this paper we will solve the CVRP by using CSO algorithm and Tabu Search (TS) separately then we create a hybrid algorithm between CSO and TS which give us the advantage of CSO in the chickens' various movements can be lead to achieve a good balance between randomness and determinism for finding the optimum, the second advantages is 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 [6]. Also we use the power of TS to avoided local optima where all the neighboring solutions are non-improving [7]. By using previous advantages for CSO and TS we will get good results that surpass the current best known results.

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 CSO and TS to solve the CVRP then the proposed hybrid algorithm to solve the problem will be discussed. The experimental results and the comparing between our hybrid algorithm with CSO, TS, and best known

results and which algorithm is better will be mentioned in sections 5.

2. Problem's formulation

The VRP is a kind of optimization problems which is the ground set is the lines of a graph $G(V,E)$. Among the different variants of VRP, the CVRP is the most essential and widespread transportation model which as Li [8] suggested, this kind of problem is easy to understand and hard to solve. The customers, exactly one depot and vehicles with the same size make the main elements for CVRP. The target of CVRP is to set paths which is beginning and ending in the depot after serving all customers under next conditions.

- Every vehicle serves a group of customers, just one time for each customer in this group.
- In each group client requirements should not override the storage space for each vehicle.
- When designing roads, we take in our consideration that the cost of the road should be minimal.

We can define the CVRP as an unguided graph $H=(X,D)$ where $X = \{x_0, x_1, \dots, x_n\}$ is a vertex set and $D = \{(x_i, x_j) / x_i, x_j \in X, i < j\}$ is an edge set. Vertex x_0 indicates for the central warehouse, and it is from where m correspondent vehicles of capacity should be able to serve all clients, dealing with a group of n vertices $\{x_1, \dots, x_n\}$. We determine on D (a non-negative cost) distance matrix $C=(C_{ij})$ between customers x_i and x_j . Let X_1, \dots, X_m be a partition of X , a route R_i is a permutation of the customers in X_i assigning the order of visiting them, starting and finishing at the depot x_0 . The cost of a given route $R_i = \{x_0, x_{i_1}, x_{i_2}, \dots, x_{i_{k-1}}, x_0\}$, where $x_{i_j} \in X$ and $x_{i_0} = x_{i_{k+1}} = x_0$ (0 denotes the depot) [1,3]. Is given by:

$$Cost(R_i) = \sum_{j=0}^{k-1} c_{j,j+1} \quad (1)$$

And the problem solution cost (S) is:

$$F_{CVRP}(S) = \sum_{i=1}^m Cost(R_i) \quad (2)$$

The CVRP consists of defining a group of m vehicle routes:

- The smallest total cost.
- Starting and stopping at the warehouse x_0 .

- Every client is visited just once by a specific vehicle, according to the limitations.
- Any route doesn't exceed the total requirements:

$$Q(\sum_{x_j \in R_i} q_j < Q) \quad (3)$$

- The total distance of any route is not larger than a pre-set specific

$$T(\text{Cost}(R_i) \leq T) \quad (4)$$

- The type of goods should be the same kind for all clients [1,3].

One of the important elements that effect on the solution is the number of vehicles as a value or variable input decision. In this research, the route length is reduced separately of the number of used vehicles, we can observe that the Travelling Salesman Problem (TSP) and VRP are nearly relevant to two combinatorial difficult problems [1]. There are a numerous examination works distributed in the VRP with pickup and delivery. Laporte has published research and studies related to the VRP and its types [9], which can be considered as a reference for this kind of problems. Pop has published a mathematical model for the popularize VRP [10].

3. Literature review

The multi-ant colony system (MACS) was used by Abad and Gajpal [11] to resolve VRP by using backhauls. There is a method to solve VRP with pickup and delivery was created by Zachariadis [12] by using an adjusted memory. For an identical issue, Subramanian [13] introduce a symmetric heuristic for solving VRPSPD which was generated to get better outcomes. Multiple theories have been introduced to solve VRP, they can be divided into exact methods, heuristic algorithms, and metaheuristic algorithms. Exact algorithms can solve small and medium VRP instances and because the limitation of exact algorithms, most algorithms used to solve VRP are heuristic and metaheuristic. Both heuristic and metaheuristic algorithms introduce approximate solutions in reasonable computing times, so they are more suitable for real-world cases and commercial applications [1]. The optimization techniques involve meta-heuristics such that, Genetic Algorithm (GA), Ant Colony System (AS), Multi-Particle Swarm Optimization (MAPSO), and Tabu Search (TS). Particularly, DVRP can be considered as an expansion to the standard VRPs that is generated by Decomposing a DVRP into a series of static VRPs. The Researchers deal

with two methods of solving the problem: Population based metaheuristics (As Ant Colony, Evolutionary Algorithms, and Particle Swarm Optimization) and trajectory-based metaheuristics (As, Tabu Search, Greedy Randomized Adaptive Search Procedure (GRASP), variable neighbourhood search). Also, a number of real-life applications and dynamic performances measures of various metaheuristics are described. The approaches based on dynamic programming could be also classified as trajectory based simulations. In DVRP is solved using an enhanced Genetic Algorithm (GA) that tries to increase both diversity and the capability to escape from local optima [14]. In the previous works and researches, a single depot with customers distributed around this depot is assumed. The route that was specified for a vehicle, start and end in a center 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 [15].

3.1 Chicken swarm optimization (CSO)

Optimization algorithm is recent bio inspired, precisely Meng was supposed in 2014 Chicken Swarm Optimization (CSO), the behaviour of the chicken swarm, the hierarchal order was mimicked. The chicken swarm can be split into many groups, every of which formed of one rooster and many hens and chicks. Different chickens follow different laws of motion. There exists contest between various chickens under certain hierarchical order [6,16]. According to the natural attitude of chickens in the swarm, the CSO introduces a swarm optimization. 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 [17]. The rooster with the highest fitness value can search for food in more places and on a larger scale.

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

where x_{ij}^{t+1} and x_{ij}^t are the position of j th dimension of particle i in $t+1$ and t iterations, respectively, and $\text{randn}(0, \sigma)$ is a random number of Gaussian distribution whose variance is σ^2 . The parameter σ^2 can be calculated [18].

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

Where $i, j \in [1, rsize]$ and $i \neq j$. $rsize$ 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 [18].

Some hens can rob good food from another group

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

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 [18].

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

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

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 [18].

Chicks search for food beside their mothers

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

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 ($m \in [1, N]$) [18].

3.2 Tabu search (TS)

Glover was proposed the Tabu search in 1986, using local search methods to defeat local optima. (Actually, Glover was suggests in 1977 many elements of this first TS and some elements of later TS proposals, specification, including short term memory to stop the reflex of current moves, and longer term frequency memory to support

- 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

attractive components). The main idea of TS is to follow up LS whenever it observe a local optimum by permit non-improving moves, cycling back to previously moves is denied by the use of memories, surname Tabu lists, that save the current history of the search, the conception that can be Subscribed to Artificial Intelligence principles .The main Tabu Search can be assuming as simply the gathering of Local Search with short-term memories. The first two basic elements of any Tabu Search heuristic are the clarification the search space and its neighbourhood. The search space of a Tabu Search or Local Search heuristic is merely the all probable solutions in the space that can be considered during the search. Orderly to give a generic template for Tabu Search, we consider that we are seek to minimize a function $f(G)$ over some space and we stratify the supposed "best improvement" version of Tabu Search as the version in which one picks at each iteration the best accessible move (this is the most normally use variant of TS).

- Notation

- G the recent solution.
- G^* the best known solution.
- f^* value of G^* .
- $N(G)$ the neighbourhood of G .
- $\tilde{N}(G)$ the allowable subset $N(G)$.

- Initialization

- Create a primarily solution:
- Put $G := G_0, f^* := f(G_0), G^* := G_0, L := \emptyset$.

- Search

- While stopping criterion not convinced do
 - Choose G in $arg \min [f(G')]; G' \in \tilde{N}(G)$
 - If $f(G) < f^*$, then set $f^* := f(G), G^* := G$;

- Save Tabu for the recent move in L (delete elderly entry if needful);
- End while

- Termination criterion

Theoretically, the search may keep going ever, except if the optimal solution of the problem is defined in advance. In pursuit, the search has to be stopped at several points. The most generally used stopping criteria in Tabu Search are:

- After a specific number of iterations (or a constant amount of CPU time).
- If there are no improvements in the objective function value after some number of iterations.
- When the objective function arrives a pre-specific threshold value. [7,19,20].

4. Proposed work

CVRP will be solved by using two algorithms the first algorithm is CSO and the second one is new hybrid CSO with Tabu Search which will be listed in the next sections. First we will describe the CSO for CVRP then the TS for CVRP and finally the hybrid algorithm for CVRP.

4.1 Solving CVRP by CSO

CVRP solved by CSO, first the parameters of CVRP like the dimension of the problem, the number of customers, the demand and coordinate of each customer, the number and the capacity of vehicles, the coordinate of the depot will be identified, also we need to specify the parameters of CSO like the number of generations and the number of iterations in each generation. The number of chickens related to the number of solutions. The initial population and first solution is generated randomly. The solution is shown by distributing customers to vehicles in Fig. 3, and in Fig. 4 the solution is clarified by clarifying the paths that the vehicles will travel in addition to arranging the customers inside the path. For more description we will take an example of for the first step is also described in Fig. 3. We can observe that the client 1 is assigned to vehicle 2, client 2 is assigned to vehicle 1, and so on. Also we have an example for second stage shown in Fig. 4. In the example, $|R1|=3$ means that the first vehicle has to serve three clients, and $j_{1,1} = 2$ means that the client 2 will be visited first by the vehicle 1. The hypothetical

solution string indicates that three vehicle routes are $0 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 0$, $0 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 0$ and $0 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rightarrow 0$ respectively. In the Fig. 3, The position X_i of a selected chicken i is the solution, $V = \{v1, v2, \dots, vn\}$. $X_i = [x1, x2, \dots, xn]$ where $(i \in [1, \dots, n])$. The fitness function is calculated for each solution. Solution will be ranked into grouped as roosters, hens, and chicks. The best solution is the first rooster then update rooster until reached to maximum number of iterations.

4.2 Solving CVRP by Tabu search

In this section we will specify the solving CVRP by using Tabu Search. First we will get the parameter of the problem like number and position of customers and the number of vehicles and the capacity of each vehicle then in the main execution step we will distribute the vehicles on the customers as per the conditions of CVRP then get the initial solution and calculate the fitness for this solution and finally the Tabu will be applied. This procedure was completed until the stopping criterion of as far as possible was come to. It depends on replacing the first customer from the first route with the place of the second customer of the first route. This solution was saved in Tabu List to avoid redundancy for the solutions and the objective function value of this solution was saved. The Fig. 6 describes the whole algorithm as a pseudo-code. It should be noted that Remaining Vehicles is considered as RV, Remaining Customers is considered as RC, Remaining Feasible Customers is denoted as RFC and Objective Function Value is denoted as OFV.

Customer no.(cno)	$c1$	$c2$	$c3$	cn
Vehicle no. (XV)	$xv1$	$xv2$	$xv3$	xvn

Customer no.(cno)	1	2	3	4	5	6	7	8	9
Vehicle no. (XV)	2	1	3	1	2	3	3	1	2

Figure. 3 Cluster – solution representation

0	$j_{1,1}$...	$j_{1,R1}$	0	$j_{2,1}$...	$j_{2,R2}$	0	...	0	$j_{v,1}$...	$j_{v,Rv}$	0
---	-----------	-----	------------	---	-----------	-----	------------	---	-----	---	-----------	-----	------------	---

0	2	4	8	0	1	5	9	0	3	6	7	0
---	---	---	---	---	---	---	---	---	---	---	---	---

Figure. 4 Tour-solution representation

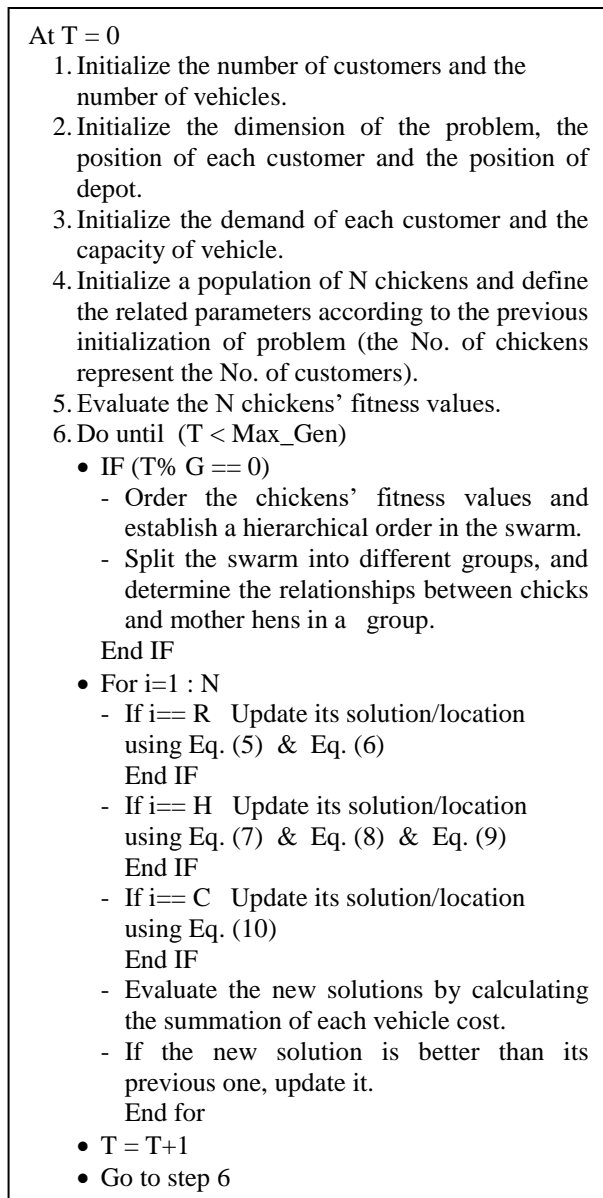


Figure. 5 Chicken swarm optimization algorithm for solving CVRP

4.3 Hybrid chicken swarm optimization with Tabu search for CVRP (HYCSOTS)

The CVRP will be solved in this part by using CSO with TS, and as we mentioned earlier, one of the advantages of using TS is that it helps to avoid local optima where all the neighboring solutions are non-improving. This advantage will be used with roosters and dominated hens to get better results. The number of clients, the number of vehicles, the demand of each customer, the capacity for the vehicles, the coordinates of the customers, and the coordinates of the depot, these parameters will be initialized for CVRP. Second, the number of iterations, populations in the swarm, also the number of roosters, hens, and chicks in the population and flow mother constant will be

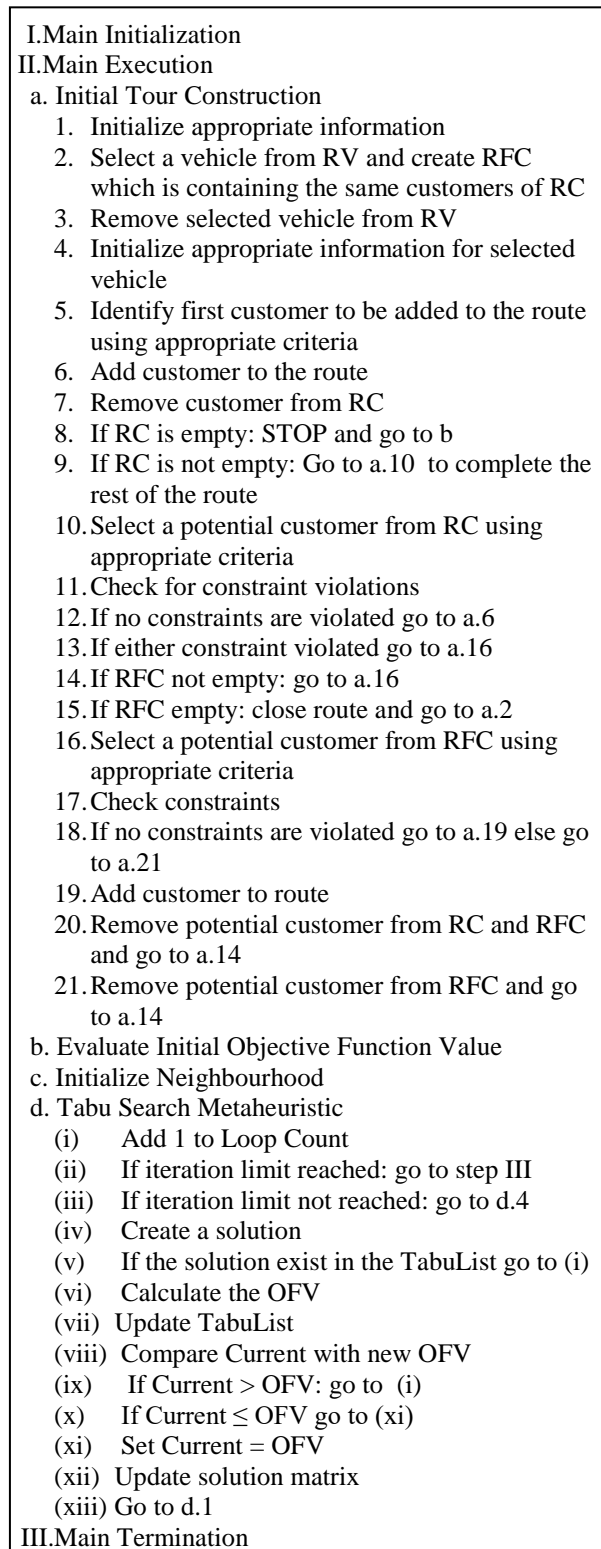


Figure. 6 The pseudo-code for solving CVRP by using TS

determined for HYCSOTS as shown in Table 2. It should be noted that the population size divided into roosters, hens and chicks, and that the more the number of iterations the more good solutions are reached. The next step is to assign every chicken to customer randomly. After that we create the initial

solution which is depends on the rules and equations of the CVRP. The most important element in creating the initial solution is that the customer requirements do not exceed the capacity of the vehicles and are calculated by the following equation: $Q(\sum_{x_j \in R_i} q_j < Q)$, then we calculate the cost of each route using the following equation: $Cost(R_i) = \sum_{j=0}^k c_{jj+1}$. We must take in the consideration the total distance of any route is not larger than a preset specific. Then we calculate the total cost for the problem by using: $F_{CVRP}(S) = \sum_{i=1}^m Cost(R_i)$. After that we apply the Tabu Search and calculate the fitness, then 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, dominated hens, mother hens and chicks. We will apply CSO moving equations on the current distribution to get I solutions, if the chicken is rooster then we will use the next equation: $x_{ij}^{t+1} = x_{ij}^t \times (1 + Randn(0, \sigma^2))$, if the chicken is hen we will use $x_{ij}^{t+1} = x_{ij}^t + S1 \times Rand \times (x_{r1,j}^t - x_{ij}^t) + S2 \times Rand \times (x_{r2,j}^t - x_{ij}^t)$, and if it is chicks we will use this equation: $x_{ij}^{t+1} = x_{ij}^t + FL \times (x_{m,j}^t - x_{ij}^t)$ to take the next step. In the other hand, we apply the Tabu Search on the same distribution but just only for roosters and dominated hens to get I' solutions, then we calculate the fitness value for every solution and compare the results and choose best fitness value and so on. The algorithm is described as follows in Fig. 7. It should be noted that we will refer to solutions obtained through CSO for roosters CR, for hens CH and we will refer to solutions obtained through Tabu Search for roosters TR, for hens TH. In the final we will consider the best solution that was got from the comparison between CR and TR in SR and for hens SH, and we will consider the solution that was got from chicks as SC.

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 Chicken Swarm Optimization (CSO), Hybrid CSO with Tabu Search (HYCSOTS) and Tabu Search (TS) results for solving CVRP also provided. The comparison is made upon the results of experiments applied on

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 I solutions of (Ni)
5. Evaluate the N chickens' fitness values
6. Improve the solution using Tabu search.
7. Set or update The N chicken's fitness values if it is better than the randomly one.
8. 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 $i=1 : N$
 - IF $i == R$ Update its solution/location using Eq. (5) & Eq. (6) to get CR End IF
 - IF $i == H$ Update its solution/location using Eq. (7) & Eq. (8) & Eq. (9) to get CH End IF
 - Use Tabu search only on roosters and dominated hens to get TR or TH.
 - Calculate the fitness value for (CR, TR) or (CH, TH) according to i .
 - IF $i == R$ and $CR \geq TR$ then $SR = CR$ else $SR = TR$
 - IF $i == H$ and $CH \geq TH$ then $SH = CH$ else $SH = TH$
 - IF $i == C$ Update its solution/location using Eq. (10) to get SC
 - End IF
 - Set or update The N chicken's fitness values if it is better than the previous one.
 - Evaluate the new solutions by calculating the summation of each vehicle cost.
 - If the new solution is better than its previous one update it
- End for
- ❖ $T = T+1$
- ❖ Go to step 8

Figure. 7 Hybrid CSO with TS for CVRP

best-known benchmarks [5]. Our methodology is implemented using the following technologies.

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

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
CSO	
Roosters Percent	15 %
Hens Percentage	70 %
Chicks percentage	15%

Average Cost

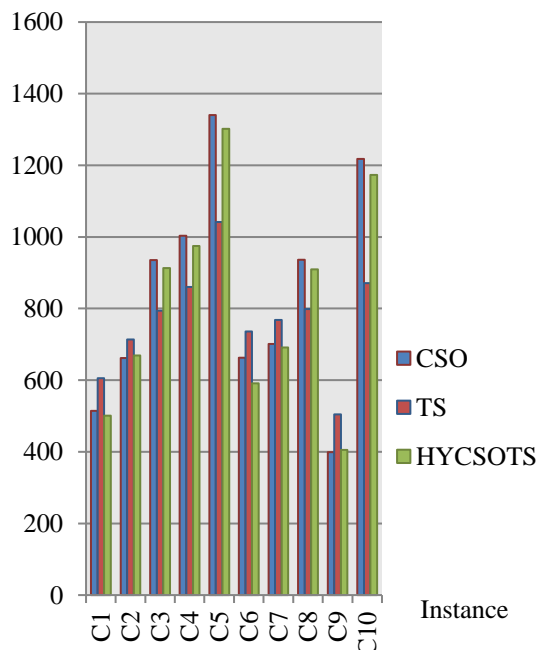


Figure. 8 Average costs for HYCSOTS, CSO and TS

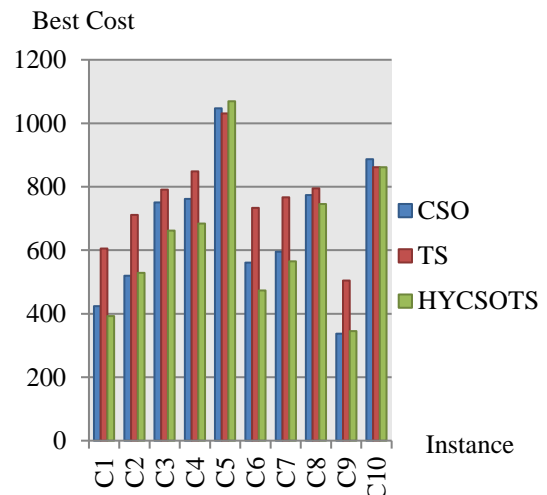


Figure. 9 Best costs for HYCSOTS, CSO and TS

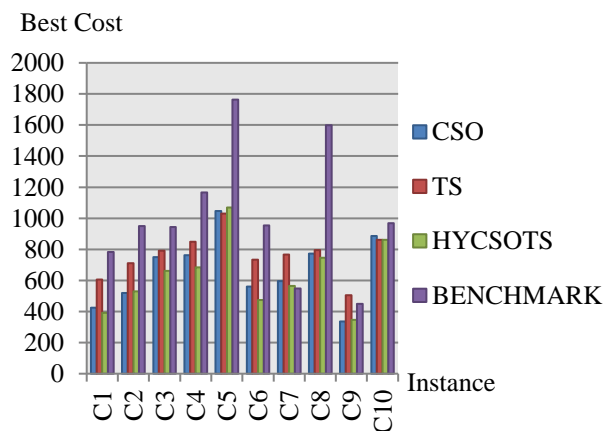


Figure. 10 Best costs for benchmark and hybrid CSO algorithms (less is better)

Table 3. Comparison between HYCSOTS, CSO, TS, and benchmark

Instance Symbol	Algorithm	Mean Cost	Maximum Cost	Minimum Cost	Standard Deviation	Hit rate %	Best Known (Benchmark)
C1	CSO	514.604359	594.6326	423.68	37.65053	100%	784
	TS	605.664521	614.0797	605.1685	1.59903868	100%	
	HYCSOTS	500.9375	597.3252	392.2502	45.02117	100%	
C2	CSO	662.0233	845.8991	519.5303	57.8781	100%	949
	TS	713.95175	733.0292	710.8494	4.457472	100%	
	HYCSOTS	668.9517	845.227	528.9327	50.05102	100%	
C3	CSO	935.632	1306.743	750.6594	92.86331	56%	944
	TS	794.467207	805.274	790.9004	4.486827	100%	
	HYCSOTS	913.2391	1184.359	661.6576	95.34	65%	
C4	CSO	1003.5489	1314.438	761.9232	98.07118	94%	1167
	TS	860.394063	892.2747	848.6098	10.12557	100%	
	HYCSOTS	974.62643	1232.438	683.6301	83.87995	98%	
C5	CSO	1340.357	1826.731	1046.661	153.8499	98%	1763
	TS	1041.773281	1082.122	1030.969	12.64363	100%	
	HYCSOTS	1301.755	1869.989	1069.21	145.8932	99%	
C6	CSO	662.569199	758.0507	560.7559	42.98174	100%	955
	TS	735.837045	746.9812	733.6974	3.504027	100%	
	HYCSOTS	590.956	688.2906	473.1573	40.65164	100%	
C7	CSO	701.655	793.5782	594.9109	44.74526	0	549
	TS	768.569471	782.9341	766.383	3.15102	0	
	HYCSOTS	691.0228	791.1641	565.4742	47.18505	0	
C8	CSO	936.621093	1179.3586	773.5207	77.49575	100%	1598
	TS	798.359891	808.512	794.9312	3.349527	100%	
	HYCSOTS	909.6509	1113.548	745.9398	76.68108	100%	
C9	CSO	398.96339	483.3567	336.925	37.76741	88%	450
	TS	504.741	504.741	504.741	0.0	0	
	HYCSOTS	405.06565	487.611	345.4381	36.99339	84%	
C10	CSO	1217.715	1613.475	886.7962	165.5583	5%	968
	TS	871.370155	904.7614	861.418	10.62801625	100%	
	HYCSOTS	1173.476	1705.321	861.8525	154.3972	7%	

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 implementation of CSO, TS and HYCSOTS algorithms on datasets using three different classes A, B and P of CVRP with instances per class. 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 HYCSOTS, CSO and TS and we consider the cost factor as will be indicated in Table 3, Figs. 8 and 9.

As the Table 3 and Figs. 8 and 9 which explains that we solve CVRP by a new hybrid CSO with TS algorithm which is solved completely and we found that the cost of the HYCSOTS algorithm is much better than CSO and TS algorithms. We indicate to solve a minimization problem so, in our graphs less results is better. The best results that exist on the benchmark [5] which were taken in 'November-2019' and our obtained results were compared.

Also as shown in Table 3 and Fig. 10 the minimum cost of Hybrid CSO with TS Algorithm is often better than benchmark cost and the percentage

of how many times that the Hybrid, CSO and TS results overcome the benchmark results through 100 runs represented as Hit rate, and our results have overcome the best results known (benchmark) completely. We found that the hybrid algorithm is better than CSO and TS and benchmark results in 5 instances (C1, C3, C4, C6, C8). For CSO is better than other results in two instances (C2, C9). The TS is better than other results in two instances (C5, C10) and finally the benchmark results are better than others in instance C7. By the comparison between the Hybrid, CSO and TS we found that the hybrid is better than others in six instances (C1, C3, C4, C6, C7, C8) also the comparison between results for CSO and TS likes the previous results. Finally we found that hybrid algorithm overcome the other two algorithms in most instances.

6. Conclusion

The CVRP is an integer programming problem, which is fall into the category of non-polynomial time problem with the goal of minimizing the total travelled distance by vehicles with limited capacity to reach their destination and back to the start depot. In this paper we use the advantages of CSO in fast convergence, easy to calculate and good robustness, consequently, achieve a good balance between randomness and determinism for finding the optimum and the advantage of TS is to avoid local optima where all the neighboring solutions are non-improving. All previous features of the two algorithms were combined to create a new hybrid algorithm HYCSOTS to solve the CVRP. From the simulation results we found that the proposed hybrid algorithm overcome the benchmark results by 90% and overcome the CSO results by 70% and overcome the TS results by 80 % which is mean the proposed hybrid algorithm almost overcome the results of benchmark, CSO and TS for CVRP. We will enhancement our proposed algorithm by using a metaheuristic algorithm like genetic algorithm or Particle Swarm Algorithm with CSO. They are generally used to generate high-quality solutions for optimization problems and search problems.

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. Willmer Escobar, "Reoptimization Heuristic for the Capacitated Vehicle Routing Problem", *Journal of Advanced Transportation*, Vol. 2018, No. 3743710, pp. 1-2, 2018.
- [2] G. I. Sayed, G. Khoriba, and M. H. Haggag, "Hybrid Quantum Salp Swarm Algorithm for Contrast Enhancement of Natural Images", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 6, pp. 225-235, 2019.
- [3] E. Alba and B. Dorronsoro, *Cellular Genetic Algorithms*, Vol. 42, Springer, 2008.
- [4] H. Y. Kang and A. Lee, "An Enhanced Approach for the Multiple Vehicle Routing Problem with Heterogeneous Vehicles and a Soft Time Window", *Symmetry*, Vol. 2018, No. 10110650, pp. 2-4, 2018.
- [5] <http://neo.lcc.uma.es/vrp/vehicle-routing-problem/>, 1-November-2019.
- [6] 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.
- [7] F. Glover and G.A. Kochenberger, *Handbook of Metaheuristics*, Vol.57, Springer, 2003.
- [8] 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.
- [9] G. Laporte, "The Vehicle Routing Problem: An overview of exact and approximate algorithms", *European Journal of Operational Research*, Vol. 59, No. 3, pp. 345-358, 1992.
- [10] C. Pop, I. Kara, and A. Marc, "New mathematical models of the generalized vehicle routing problem and extensions", *Applied Mathematical Modelling*, Vol. 36, No. 1, pp. 97-107, 2012.
- [11] Y. Gajpal and P.L. Abad, "Multi-ant Colony System (MACS) for a Vehicle Routing Problem with Backhauls", *European Journal of Operational Research*, Vol. 196, No. 1, pp. 102–117, 2009.

- [12] E. Zachariadis, D. Tarantilis, and T. Kiranoudis, "A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service", *Expert Systems with Applications*, Vol. 36, No. 2, pp. 1070–1081, 2010.
- [13] A. Subramanian, L.M.A. Drummonda, C. Bentes, L. S. Ochi, and R. Farias, "A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery", *Computers & Operations Research*, Vol. 37, No. 11, pp. 1899–1911, 2010.
- [14] E. Kucharska, "Dynamic Vehicle Routing Problem—Predictive and Unexpected Customer Availability", *Symmetry*, Vol. 11, No. 546, pp. 2, 2019.
- [15] 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.
- [16] C. Qu, S. Zhao, Y. Fu, and W. He, "Chicken Swarm Optimization Based on Elite Opposition-Based Learning", *Mathematical Problems in Engineering*, Vol. 2017, No. 2734362, pp. 1-20, 2017.
- [17] 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.
- [18] 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.
- [19] F. Montané and R.D. Galvão, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", *Computers & Operations Research*, Vol. 33, No. 3, pp. 595–619, 2006 .
- [20] A. Toshev, "Particle Swarm Optimization and Tabu Search Hybrid Algorithm for Flexible Job Shop Scheduling Problem – Analysis of Test Results", *Cybernetics and Information Technologies*, Vol. 19, No. 4, pp. 33-39, 2019.