Hindawi
Advances in Civil Engineering
Volume 2023, Article ID 2728039, 15 pages
https://doi.org/10.1155/2023/2728039

Research Article

Cement Transport Vehicle Routing with a Hybrid Sine Cosine Optimization Algorithm

Vu Hong Son Pham and Van Nam Nguyen

Department of Construction Engineering and Management, Ho Chi Minh City University of Technology (HCMUT), Vietnam National University (VNU-HCM), Ho Chi Minh City, Vietnam

Correspondence should be addressed to Van Nam Nguyen; nvnam.sdh20@hcmut.edu.vn

Received 19 December 2022; Revised 18 January 2023; Accepted 1 February 2023; Published 17 February 2023

Academic Editor: Ankit Gupta

This study will solve the classical vehicle routing problem, the goal is to generate k trips with the shortest distance for h customers with predetermined locations and needs.. The proposed solution to the classical vehicle routing problem is a hybrid sine cosine algorithm. The sine cosine algorithm is hybridized with the grey wolf optimizer, which is used in combination with the methods of tournament selection, opposition learning, and the mutation and crossover method to build the optimal routing plan for the means of transporting cement. To demonstrate the advantages of the developed hybrid sine cosine algorithm, this algorithm is evaluated and compared with modern algorithms such as sine cosine algorithm, dragonfly algorithm, grey wolf optimizer, ant lion optimizer, particle swarm optimization, modified hybrid particle swarm optimization, genetic algorithm, and the double-population genetic algorithm in case studies. The hybrid sine cos algorithm gives optimal results in these cases because it balances mining and exploration. Thus, the results of this study indicate that managers can use the developed hybrid sine cosine algorithm to create optimal vehicle routing plans to reduce transportation distances.

1. Introduction

The TSP problem is computationally complex [1]. In this problem, for n cities, $1/2 \times (n-1)!$ paths are possible. For example, if $n = 16$, the number of possible paths is 6.54×10^{11}; thus, the number of paths is too high in the TSP. The VRP is an extension of the TSP; thus, the VRP has considerably high computational complexity.

In the classical vehicle routing problem, which is an extended version of the traveling salesman problem, the objective is to generate a set of k trips with the shortest distance or minimum cost for h customers whose locations and demands are predetermined. Each vehicle travels to and from a fixed location and satisfies some associated constraints. The vehicle routing problem can be solved using many methods, such as linear programming, genetic algorithm (GA), ant lion optimizer (ALO), dragonfly algorithm (DA), particle swarm optimization (PSO), modified hybrid particle swarm optimization (MHPSO), and double population genetic algorithm (DPGA).

The vehicle routing problem (VRP) is a classical NP-hard problem that is difficult to not only solve but also define. According to Laporte [2], the VRP cannot be precisely defined because of the diversity and complexity of the binding requirements in this problem (e.g., time, distance, cost, pick-up and delivery, and capacity). Therefore, studies should focus on the most important parameters such as distance [3], cost [4], and time and CO_2 emissions [5] when solving the VRP. According to Liu et al. [6], the main difference between the traveling salesman problem (TSP) and VRP is that in the VRP, multiple routes can be created to pass through all nodes under the condition of limited vehicle capacity. Because of the VRP's complexity, almost all studies on this problem have attempted to solve it by using heuristic and meta-heuristic methods.

The VRP problem has attracted the attention of many authors because of its practical applicability. In fact, there cannot be a vehicle that can hold goods to deliver to customers when the number of customers is too large and the goods are too bulky in size, while it may be assumed that such a vehicle exists, it remains an unattainable assumption to deliver goods on time to customers when multiple orders are placed at the same time. We know that each vehicle will have a certain capacity and load, so it is necessary to plan the routing of delivery vehicles to meet the daily needs of customers. Regardless of the method used, the researchers want to provide the optimal solution of the objective function: cost, distance, delivery time, and CO_2 emissions, to deliver goods to customers from one repository [7, 8] or from multiple repositories [9, 10]. The issue of optimizing CO_2 emissions has been studied a lot in recent years [6, 11] because of global warming and countries starting to tax the amount of CO_2 emissions from transport vehicles.

Nowadays, optimization algorithms are applied in many different fields. In recent years optimization algorithms are being developed rapidly such as the monarch butterfly optimization (MBO) [12], slime mold algorithm (SMA) [13], moth search algorithm (MSA) [14], hunger games search (HGS) [15], Runge–Kutta method (RUN) [16], colony predation algorithm (CPA) [17], weIghted meaN oF vectOrs (INFO) [18], Harris hawks optimization (HHO) [19, 20], improvement of trajectory tracking by robot manipulator based on a new cooperative optimization algorithm [21], optimal design of low computational burden model predictive control based on SSDA towards autonomous vehicle under vision dynamics [22], improved grey wolf optimizer based on opposition and quasi learning approaches for optimization [23], effective nonlinear model predictive control scheme tuned by improved NN for robotic manipulators [24], GSA-based design of dual proportional integral load frequency controllers for nonlinear hydrothermal power system [25], development of an IoT architecture based on a deep neural network against cyber attacks for automated guided vehicles [26], reliable deep learning and IoT-based monitoring system for secure computer numerical control machines against cyber-attacks with experimental verification [27], a multistrategy-enhanced sine cosine algorithm for global optimization and constrained practical engineering problems [28], optimal allocation of distributed generators in active distribution networks using a new oppositional hybrid sine cosine muted differential evolution algorithm [29], and sine cosine algorithm with multigroup and multistrategy for solving CVRP [30].

This study combines the sine-cosine algorithm with the grey wolf optimizer, combined with tournament selection, adversarial learning, mutation crossover, and other methods, to obtain the optimal path scheme for cement transportation creation. This combined algorithm is called the hybrid SCA (HSCA). The SCA with outstanding strength in expanding the search space and GWO algorithm with outstanding mining ability is used in the search for optimal results. The TS method improves the ability to select individuals in the population to run the algorithm. The OBL method enhances global discoverability and enhances local mining through the convergence acceleration parameter, the MCS method increases the chance of finding optimal results through transformability and improves the ability to discover. From there, the HSCA algorithm will help the VRP problem achieve the best balance between the discovery and mining stages. The study will focus on the application of the HSCA algorithm to discover a routing plan for cement transport vehicles so that "the total delivery distance between the nodes is the shortest."

In this study, a new algorithm HSCA (SCA-GWO-OBL-MCS) was created and used to solve the CVRP problem of distance optimization, saving logistics costs. HSCA has proven its improvement effectively compared to other algorithms by 2 case studies presented in Chapter 4.

2. Literature Review

The VRP has been studied for more than 60 years [31, 32], with diverse objectives being adopted and diverse solutions being proposed. When considering distance and customer demands in the VRP [33], this problem can be solved using the "3-opt" model and mixed-integer linear programming (MILP) if the vehicles have the same size or different sizes, respectively [34].

Anbuudayasankar and Mohandas [3] used MILP to optimize the pick-up and delivery of commodities at 13 nodes under different distances and customer demands. Moreover, Wang et al. [35] considered travel and service times in the VRP and solved this problem using the ant colony optimization (ACO) algorithm. Qi and Hu [4] solved the VRP for the transportation of frozen commodities to 13 locations by attempting to minimize vehicle damage, fuel costs, and refrigeration costs.

Reed et al. [36] used ant colony system (ACS) to route vehicles for nodes in cyberspace and extended it for modeling the usage of multichambered vehicle to sort waste. A reasonable solution to the VRP reduced delivery costs by 15% in a management science project at E. I. Du Pont, Inc. [33]. Venkata Narasimha et al. [37] formulated a VRP that is different from the traditional min-max multidepot vehicle routing problem to reduce the travel time of the vehicle that travels the longest distance. Such a reduction is meaningful in emergency response scenarios to minimize the arrival time of any customer. In addition, some researchers have considered carbon emissions for effectively solving the VRP. Liu et al. [6] used the genetic algorithm (GA) to identify the minimal-carbon-footprinttime-dependentheterogeneous-fleet vehicle routing problem with alternative paths under different vehicle capacities and vehicle speeds (the vehicle speed changes at different delivery times during a day). Wang et al. [5] used a GA-based method to optimize a cold cargo distribution route in China for minimizing the carbon tax to be paid, travel time, and transportation cost.

A MILP model developed in 2016 by Afshar-Bakeshloo et al. [8] provided a VRP problem-solving model with heterogeneous vehicle capacity to serve a group of customers in a predetermined time period; in this model, in addition to cost optimization, emissions and customer satisfaction are also considered. In this paper [38], an accurate heuristic-based approach is developed to solve the green vehicle routing problem, which extends the classical vehicle routing problem by considering a limited range of vehicles with limited refueling infrastructure. In 2019, Wang [9] combined 3 algorithms Clarke and Wright savings heuristic algorithm, the sweep algorithm, and the multiobjective particle swarm optimization algorithm to solve the problem of routing vehicles with many different warehouses. This is a 2-objective model of CO_2 emissions and operating costs and it also implements a penalty function for early or late delivery to reduce waiting times and improve customer satisfaction. When it comes to customer satisfaction, Wang says that finding the shortest distance sometimes also falls short of goals for cost and minimum CO_2 emissions.

In 2020, Zhang proposed a [10] multidepot green vehicle routing problem, in which clean fuel vehicles start from different depots; after serving the customer, the vehicle will have to return to the original warehouse with the goal of reducing CO_2 emissions to the environment because there is not enough fuel, and the vehicle will have to go to the station to refuel; the two-stage ant colony system method is proposed to solve this problem. In addition, the VRP problem is also known in another form as the dynamic problem, Khouadjia et al. [39] used particle swarm optimization and variable neighborhood search to solve this dynamic vehicle routing problem. In the dynamic vehicle routing problem, new orders appear when the goods plan is being delivered, so the routes have to be rearranged to deliver goods to the customer while the old model is still in progress, this is also an extended problem of the traditional VRP problem.

3. Materials and Methods

Chapter 3 identifies the VRP problem and introduces the hybrid SCA methods with GWO, OLB, MCS, and TS applied to the case study in Chapter 4.

3.1. Description of VRP Problem. There are many documents on how to solve the VRP problem according to different definitions. According to Bodin's definition [40], the VRP problem is developed through the mixed-integer programming (MIP) in which the variables are integers associated between the arcs of the location. This model is also called as vehicle flow model. Another quite effective way of describing the capacity problem of VRP is introduced by Shan and Wang [41] which is described as a summation model for the problems and optimization is implemented by the particle swarm optimization (PSO) which is quite effective.

The CVRP problem is specifically defined by Shan and Wang [41] as follows:

It is assumed that there is a warehouse and vehicles will transport commodity to predetermined customers from the warehouse, and after delivery, the vehicles will return to the warehouse. And the additional constraints are given as follows:

- Cargo vehicles are limited in capacity
- Each customer comes to deliver the commodity only one time

Objective: to optimize the delivery distance of many vehicles at the same time.

Variables:

$$D = \text{total distance travelled by all vehicles},$$
$$x_{ijs} = \left\{ \begin{array}{c} 1, \text{vehicle } s \text{ depart from } i \ to \ j \\ 0, \text{otherwise} \end{array} \right\}, \tag{1}$$
$$y_{is} = \left\{ \begin{array}{c} 1, \text{customer } i \text{ is served by vehicle } s \\ 0, \text{otherwise} \end{array} \right\}.$$

Coefficients:

$$\begin{aligned} c_{ij} &= \text{cost form customer } i \text{ to customer } j, \\ g_i &= \text{the de mand of th } i\text{th customer } (i = 1, 2, 3 \ldots, h), \\ h &= \text{total number of customer}, \\ k &= \text{total number of vehicle}, \\ q_s &= \text{capacity of vehicle s}, \\ s &= \text{the.vehicle.number } (i = 1, 2, 3 \ldots, k). \end{aligned} \tag{2}$$

Objective function:

$$\text{Min } D = \sum_{i=o}^{h} \sum_{j=0}^{h} \sum_{s=1}^{k} c_{ij} x_{ijs}. \tag{3}$$

Constraints:

$$\sum_{i=0}^{h} x_{ijs} = y_{js}, j = 1, 2, \ldots, h, \quad s = 1, 2, \cdots, k, \tag{4}$$

$$\sum_{i=0}^{h} x_{ijs} = y_{is}, j = 1, 2, \ldots, h, \quad s = 1, 2, \ldots, k, \tag{5}$$

$$\sum_{i=0}^{h} g_i y_{is} \leq q_s y_{is}, s = 1, 2, \ldots, k, \tag{6}$$

$$\sum_{s=1}^{k} y_{is} = \left\{ \begin{array}{c} 1, i = 1, 2, 3, \ldots, h \\ k, i = 0 \end{array} \right\}. \tag{7}$$

Equation (3) presents the VRP's objective function. In this equation, x_{ijs} is a binary variable that indicates whether the current path has been selected. Equations (4) and (5) indicate that only one path exists between a vehicle and a specific customer. Equation (6) incorporates vehicle capacity as a limitation. Equation (7) indicates that each customer is served by only one vehicle and that the warehouse is served by k vehicles.

3.2. Introduction of SCA. The SCA, which was introduced by Mirjalili [42], is a population-based optimization method. The SCA generates many initial choices and allows them to fluctuate toward the best solution by using the sine and cosine functions.

The optimization process of population-based optimization algorithms is divided into two phases: exploitation and exploration [42].

The equation represents the two phases of exploitation and exploration follows:

$$X_i^{t+1} = X_i^t + r_1 * \sin(r_2) * \left|r_3 * P - X_i^t\right|, \quad (8)$$

$$X_i^{t+1} = X_i^t + r_1 * \cos(r_2) * \left|r_3 * P - X_i^t\right|. \quad (9)$$

The two formulas (8) and (9) are combined as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 * \sin(r_2) * \left|r_3 * P - X_i^t\right|; r_4 \leq 0.5 \\ X_i^t + r_1 * \cos(r_2) * \left|r_3 * P - X_i^t\right|; r_4 \geq 0.5 \end{cases}, \quad (10)$$

where r_4 is a random value in the interval $[0; 1]$.

The main parameters in formulas (7)–(9) are r_1,r_2,r_3, and r_4.

(i) r_1 shows different directions within the space between the solution and the destination or the space outside the solution.

(ii) r_2 determines the distance to travel outward or toward the destination.

(iii) r_3 gives random weights to emphasize the weights of the processes of exploration ($r_3 > 1$) or exploitation ($r_3 < 1$).

Effects of sine and cosine in equations (8) and (9) are illustrated in Figure 1. This figure shows how equations (8) and (9) define the spatial region between two solutions in the search space. The two-dimensional model is illustrated in Figure 1, however, the search space can be extended to higher dimensions. The cyclical model of sine and cosine functions allows repositioning around a different solution. This way certainly ensures the exploitation of the space defined between the two solutions.

In addition, in order to avoid local optimization, solutions shall be searched outside the space between the solutions and the destination. To achieve this, it is necessary to change the ranges of the sine and cosine functions shown in Figure 2.

The conceptual model of the effects of sine and cosine functions in the range (−2, 2) is illustrated in Figure 3. Figure 3 shows that changing the range of the sine and cosine functions will lead to one solution which changes the position of this solution itself with another.

For a good algorithm, a balance of the two phases of exploration and exploitation is requested to discover promising regions of the search space to reach the global optimization.

For SCA, in order to balance the exploration and exploitation, the range of sine and cosine functions in formulas (7)–(9) is changed by the following formula:

$$r_1 = a - t\frac{a}{T}, \quad (11)$$

where t is the current iteration, T is the maximum number of iterations, and a is a constant.

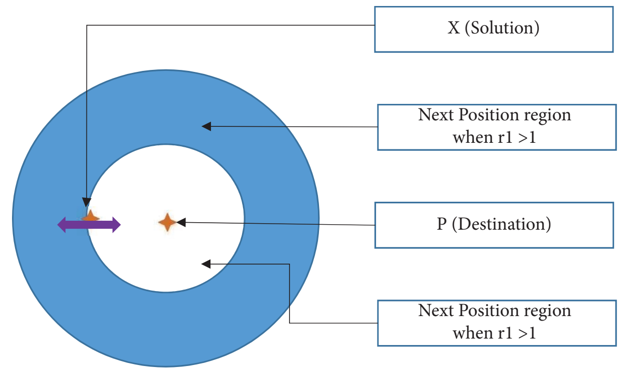

Figure 1: Effects of the sine and cosine function in updating the latest value [43].

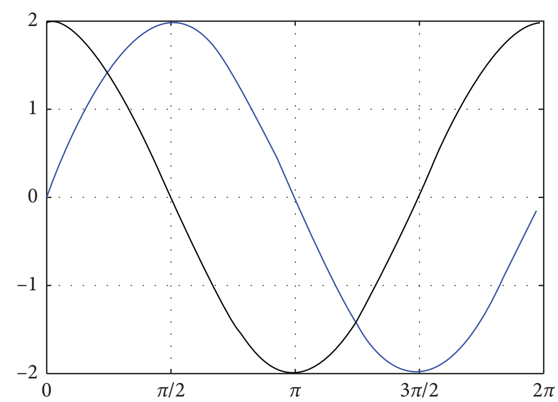

Figure 2: Effects of sine and cosine in formulas (7) and (8) on the next region and position.

Figure 4 shows that equation (11) reduces the range of sine and cosine functions over iterations. Based on Figures 3 and 4, it can be seen that the SCA algorithm shall explore the search area when the range of sine and cosine functions is in the interval $(1, 2]$ and $[-2, -1)$. However, the algorithm shall exploit the search space when the range is in the interval (−1, 1) (see Algorithm 1).

The SCA algorithm starts the optimization with a set of random solutions. The algorithm then saves the best solutions obtained so far, assigns it as the destination, and updates other solutions with respect to it. Meanwhile, the scope of the sine and cosine functions is updated to emphasize the exploitation of the search space as the iteration counter increases. The SCA algorithm terminates the optimization process when the iteration counter goes higher than the maximum number of iterations by default. However, any other termination condition can be considered such as maximum number of function evaluation or the accuracy of the global optimum obtained. The flowchart of SCA algorithm is another demonstration of Figure 5.

3.3. Introduction of OBL and MCS Methods

3.3.1. OBL Method. The concept of opposition-based learning and its applications was introduced in 2005 by Tizhoosh [44] (illustrated in Figure 6).

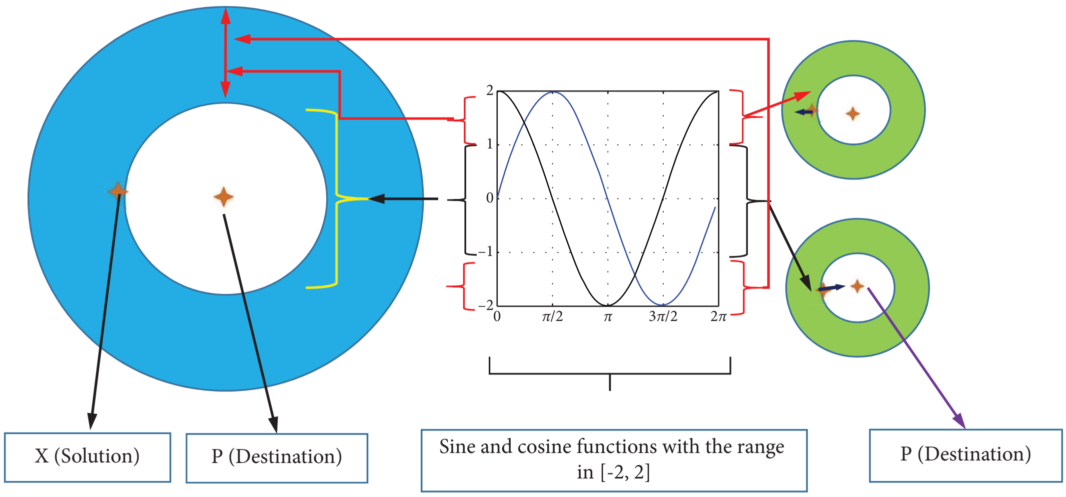

Figure 3: Effects of the function in the range (−2, 2) enabling a solution to go outside or around the destination [42, 43].

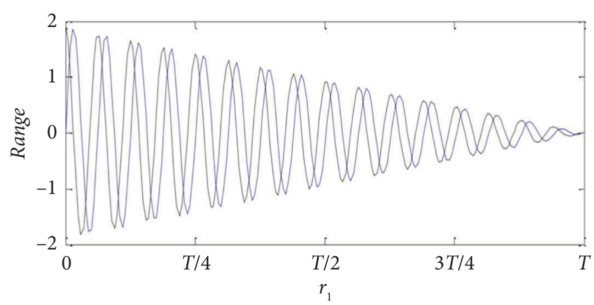

Figure 4: Model of descending the range of sine and cosine functions ($a = 3$).

Initialize a set of search agents (solutions) (X).
Do
  Evaluate each of the search agents by the objective function.
  Update the best solution obtained so far ($P = X^*$)
  Updater_1, r_2, r_3, and r_4
  Update the position of search agents using equation (10)
While ($t <$ maximum number of iterations)
Return the best solution obtained so far as the global optimum.

Algorithm 1: Steps of SCA algorithm [41].

Let $P(x_1, x_2, \ldots, x_n)$ be a point in n-dimensional space, where $x_1, x_2, \ldots, x_n \in R$ and $x_i \in [a_i, b_i] \forall i \in \{1, 2, \ldots, n\}$. The opposite point of P is defined as $\overline{P}(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n})$, where

$$\overline{x_n} = a_i + b_i - x_i. \tag{12}$$

Assume $f(x)$ is a suitable function used to measure the solution's optimum. If $f(\overline{P}) \geq f(P)$, then the point P can be replaced by $\overline{P}$, otherwise we shall continue with P.

3.4. Introduction of MCS Method

3.4.1. MCS Method. This method is introduced by Roseline and Saravanan [45] as follows:

Mutation and crossover are popular exploitation algorithms in various phases of optimization. Each $x_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ is a vector of n directions (n dimensions).

Step 1. Mutation.

The mutation algorithm can generate a mutation vector u_i by choosing random components from the directions of the original vector x_i and reassign the value in the specified domain of such direction as follows:

$$u_{ij} = \begin{cases} \text{rand}\left(x_{ij} \in \left[\text{Domainof} x_{ij}\right]\right); \text{rand} \leq p_c. \\ x_{ij}, \text{otherwise}. \end{cases} \tag{13}$$

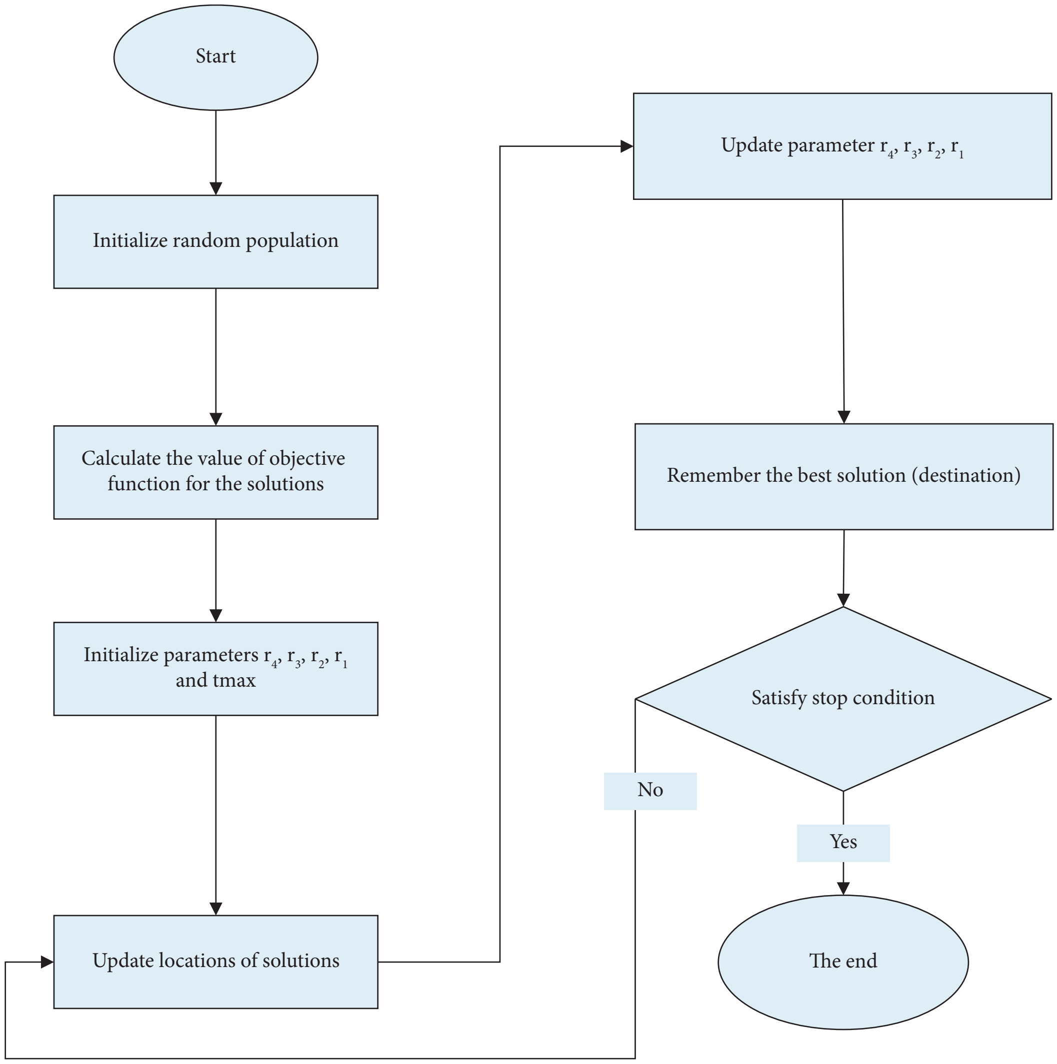

Figure 5: The flowchart of SCA algorithm.

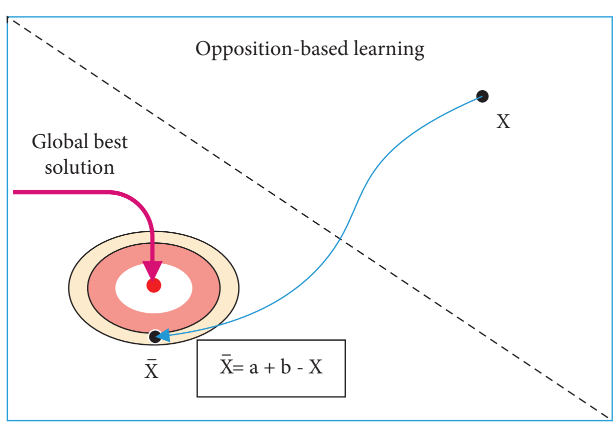

Figure 6: OBL method.

Step 2. Crossover.

The crossover algorithm can generate a new vector v_i by using a crossover with the mutation vector. New vector v_i is generated by random selection of components from vector u_i and target vector x_i based on the probability factor p_c.

$$v_{ij} = \begin{cases} u_{ij}, \text{rand} \leq p_c \text{ or } j = j_o, \\ x_{ij}, \text{otherwise,} \end{cases} \quad (14)$$

where p_c is the probability factor which controls the population diversity and reduces the risk of local optimization (set $p_c = 0.1$ in this study) and j_0 is an indicator belonging to $(1, 2, 3, \ldots, n)$ to ensure that vector v_i contains at least one element from the mutation vector u_i.

3.5. Introduction of Tournament Selection Method. In the SCA algorithm, the roulette method in the individual selection step is replaced by tournament selection. The tournament selection procedure randomly selects k elements and compares the values of the objective function. We selected elements with better objective function values. This technique helps to improve the ability to quickly find the optimal value of the SCA algorithm.

3.6. Introduction of GWO. In 2014, Mirjalili et al. [46] introduced the GWO algorithm. We approached hunting and hierarchical leadership like wolves in the wild, where there are 4 levels: alpha, beta, delta, and omega. The first three wolves will be the best variant of the population, and omega (ω) is the population variant. Wolf population has 2 stages: siege and hunt for prey.

The siege phase is displayed as follows:

$$\begin{aligned} \vec{d} &= \left| c.\vec{x}_p^{\,t} - \vec{x}^{\,t} \right|, \\ \vec{x}^{\,t+1} &= \vec{x}^{\,t} - \vec{a}.\vec{d}, \end{aligned} \quad (15)$$

where $\vec{x}\,(t)$ is the wolf's position in iteration t, $x\,(t)$ is the prey's position, and vectors $\vec{a}$ and $\vec{c}$ are coefficient vectors, computed as follows:

$$\begin{aligned} \vec{a} &= 2l.r_1, \\ \vec{c} &= 2r_2. \end{aligned} \quad (16)$$

Hunting phase: To simulate hunting behavior, Mirjalili assumes that the knowledge of the potential position of the prey is known by alpha, beta, and delta based on its experience as follows:

$$\begin{aligned} \vec{d}_\alpha &= \left|\vec{c}_1.\vec{x}_\alpha - \vec{x}\right|, \\ \vec{d}_\beta &= \left|\vec{c}_2.\vec{x}_\beta - \vec{x}\right|, \\ \vec{d}_\delta &= \left|\vec{c}_3.\vec{x}_\delta - \vec{x}\right|, \\ \vec{x}_1 &= \vec{x}_\alpha - \vec{a}_1.\left(\vec{d}_\alpha\right), \\ \vec{x}_2 &= \vec{x}_\beta - \vec{a}_2.\left(\vec{d}_\beta\right), \\ \vec{x}_3 &= \vec{x}_\delta - \vec{a}_3.\left(\vec{d}_\delta\right) \\ &\cdot \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3}, \end{aligned} \quad (17)$$

When searching and attacking prey, $\vec{a}$ is a random value in the range of $(-2a, 2a)$. When $|\vec{a}| < 1$, random prey value is attacked by wolves, it is the mining stage. When $|\vec{a}| > 1$, wolves are forced to ignore their prey in search of better prey [47].

Another parameter that affects the decoy search is that c has a value in the range (0, 2), and $\vec{c}$ will randomly and suddenly update the value of the solution to avoid local optimization. If $c > 1$, the solution converges towards the prey, and if $c < 1$, the solution moves away from the prey to find new prey.

Finally, the flowchart of the SCA-GWO-TS-OBL-MCS algorithm is shown in Figure 7.

4. Application of the HSCA to Solve the VRP

4.1. Case Study 1. The developed HSCA was evaluated against other algorithms in solving the problem of [48] (case study 1).

This problem is expressed as follows. For a central warehouse, eight customers are served by two trucks, each of which has a capacity of 8 units. The matrix of travel distances and customer demands is presented in Table 1. The requirement is that two vehicles should deliver the goods so that the shortest total delivery distance is achieved and the conditions of the VRP stated in Section 3 are satisfied.

A personal laptop with an 11th Gen Intel(R) Core(TM) i7-1165G7@ 2.80 GHz processor was used to run each algorithm 20 times with 20 search agents in 50 iterations. The HSCA was programmed in Java. The shortest distance obtained was 67.5 units, and the following paths were obtained in Java for the two vehicles (in the output order in Java):

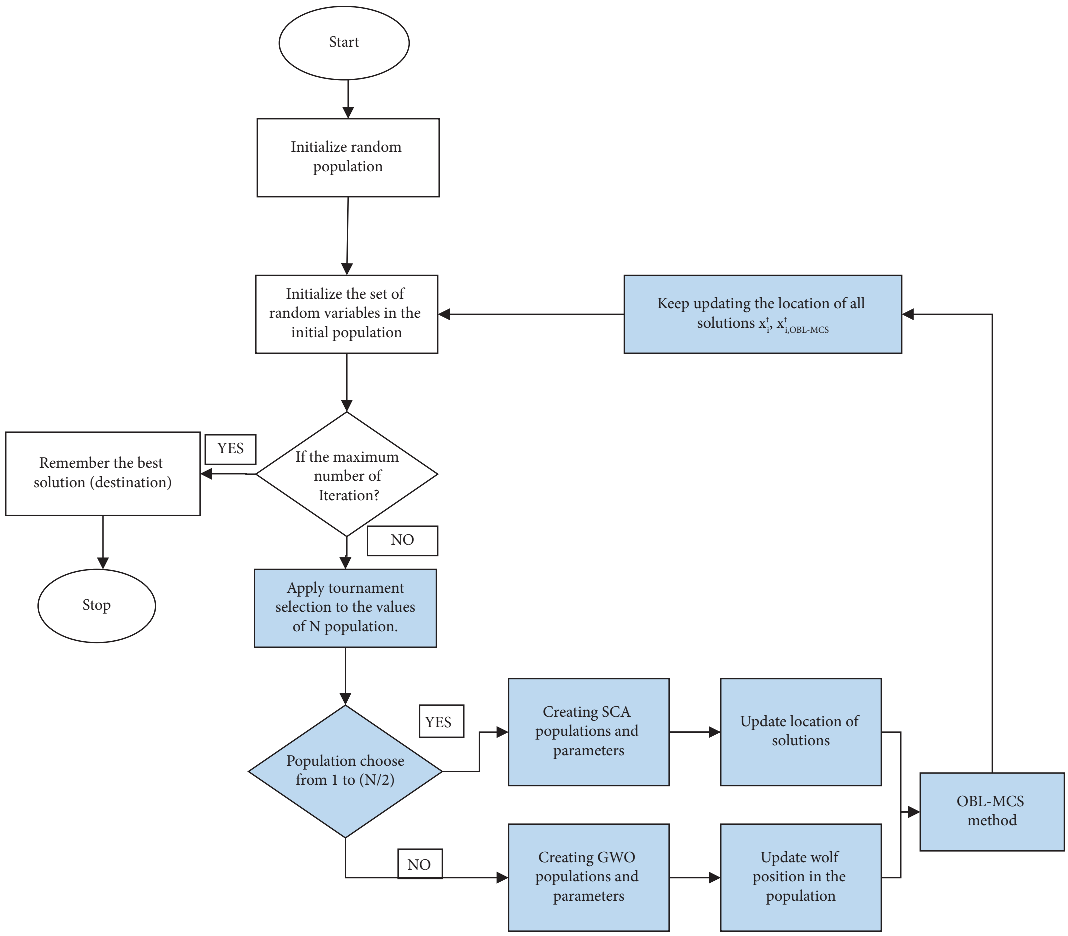

Figure 7: The flowchart of HSCA.

Table 1: Matrix of travel distances and customer demands case study 1 [48].

Customer	0	1	2	3	4	5	6	7	8	Demand
0	0	4	6	7.5	9	20	10	16	8	
1	4	0	6.5	4	10	5	7.5	11	10	1
2	6	6.5	0	7.5	10	10	7.5	7.5	7.5	2
3	7.5	4	7.5	0	10	5	9	9	15	1
4	9	10	10	10	0	10	7.5	7.5	10	2
5	20	5	10	5	10	0	7	9	7.5	1
6	10	7.5	7.5	9	7.5	7	0	7	10	4
7	16	11	7.5	6	7.5	9	7	0	10	2
8	8	10	7.5	15	10	7.5	10	10	0	2

Vehicle 1: 0 -> Customer6 -> Customer7 -> Customer 4 -> 0.

Vehicle 2: 0 -> Customer1 -> Customer3 -> Customer5 -> Customer8 -> Customer 2 -> 0.

Figure 8 depicts the HSCA's optimal solution, and Table 2 presents the smallest number of iterations required for obtaining the optimal result with different algorithms. The aforementioned figure and table indicate that the HSCA can search for optimum solutions faster and more efficiently than the other algorithms used in this study.

The results obtained with the HSCA, SCA, DA, PSO, and ALO were compared. The results indicated that the HSCA achieved convergence faster than the other four algorithms. Figure 9 displays the average best solutions obtained with the aforementioned five algorithms after 50 iterations and 20 runs.

Table 3 presents the results obtained in case study 1 when using different algorithms. The average distance value obtained with the HSCA (i.e., 67.675 units) was lower than those obtained with the other algorithms.

The results obtained with the HSCA were more stable (i.e., had a smaller deviation) than those obtained with the other algorithms, and the global optimal solution was obtained with the HSCA. Thus, the HSCA can be used for cement vehicle routing. Cement vehicles have similar

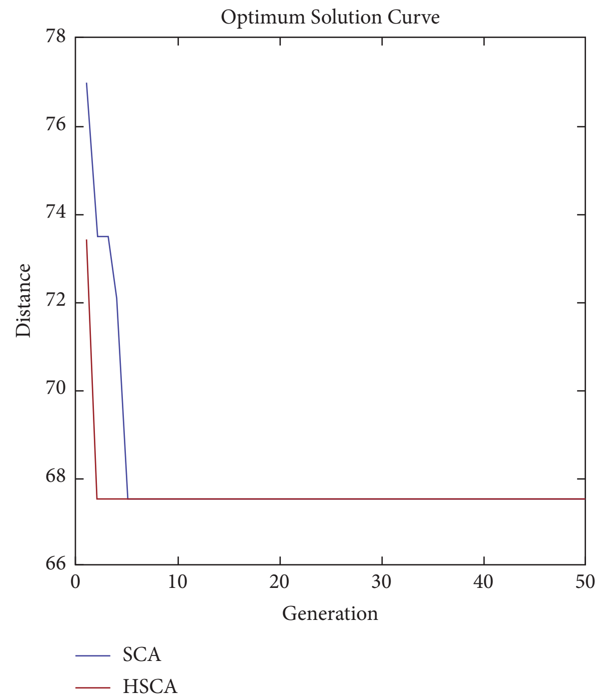

Figure 8: Optimal values obtained by the HSCA and SCA after 20 runs in case study 1.

Table 2: Optimal results obtained with different algorithms after 20 runs in case study 1.

	HSCA	SCA	DA	PSO	ALO	MHPSO* [48]	DPGA* [48]	SGA* [48]
Generation	2	5	9	12	8	6	18	45

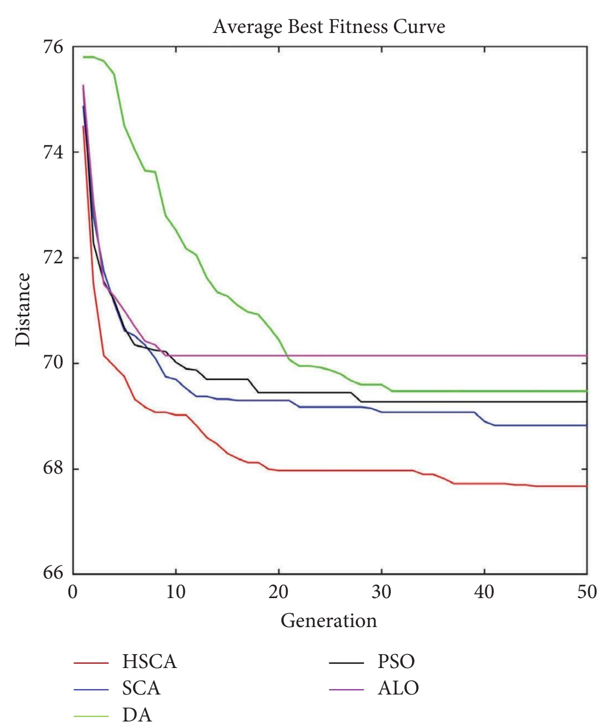

Figure 9: Optimal average values were obtained with different algorithms after 20 runs in case study 1.

Table 3: Best solutions obtained in case study 1 with different algorithms after 20 runs.

	Distribution of optimum solutions					Max	Min	Mean
HSCA	67.5	68	67.5	67.5	67.5	**69**	**67.5**	**67.675**
	68.5	67.5	67.5	67.5	67.5			
	67.5	68	68	68	68			
	67.5	67.5	67.5	67.5	67.5			
SCA	69	68	69	68	68	69	67.5	68.025
	68	69.5	67.5	67.5	68			
	69	68	67.5	67.5	67.5			
	68	67.5	68	67.5	67.5			
DA	71.5	67.5	71.5	68	67.5	71.5	67.5	68.725
	69	70	70.5	68	69			
	70	67.5	67.5	69	68			
	67.5	68	68	67.5	69			
ALO	71.5	68	71.5	68	67.5	71.5	67.5	69.150
	69	70	70.5	68	69			
	70	68	71	69	68			
	71.5	68	68	67.5	69			
MHPSO	69.5	67.5	69	69	70	70	67.5	68.875
	69.5	70	69	67.5	67.5			
	69	69.5	69	70	67.5			
	70	69	67.5	70	67.5			
PSO	67.5	70	70	69	69	70	67.5	68.950
	68	69	70	70	68.5			
	68.5	68.5	67.5	68	70			
	70	67.5	69.5	69	69.5			
DPGA	70	69	67.5	71	69	72	67.5	69.550
	70.5	72	67.5	71.5	69			
	67.5	69	71	70	67.5			
	70.5	69	69.5	71	69			
SGA	69	72	73.5	69	70	75.5	67.5	70.425
	71	67.5	69	69	75.5			
	70	69.5	69	73	69			
	74	70	69.5	69	70			

Bold values represent better HSCA search results than other algorithms.

capacities and are delivered to different warehouses and distribution stores daily. The HSCA can be used to obtain optimal routes for cement vehicles depending on the delivery case and time of day.

4.2. Case Study 2: Real Problem of Cement Delivery to Customers. The real problem of a cement distribution agent (case study 2) is expressed as follows: In 1 day, a cement distribution agent with a central warehouse must serve 30 customers by using five trucks, each of which has a maximum capacity of 90 cement bags during a single delivery. The matrix of travel distances and customer demands for case study 2 is presented in Table 4. The requirement is that five vehicles should deliver the goods, such that the shortest total delivery distance is achieved and the relevant conditions of the VRP stated in Section 3 are satisfied.

A personal laptop was used to run each algorithm 20 times with 50 search agents in 200 iterations. The HSCA was programmed in Java. The best solution for the total distance was 701.1625 units. Figure 10 displays optimal values obtained by the HSCA. The following paths were obtained in Java for the five vehicles (in the output order in Java):

A comparison of the results obtained with the DA, PSO, ALO, the SCA, and the HSCA in case study 2 indicated that the HSCA converged faster than the other algorithms. Figure 11 displays the average best solutions obtained with the different algorithms after 200 iterations and 20 runs.

Table 5 presents the maximum values, minimum values, mean values, and deviations to the best solution obtained with the different algorithms used in this study. After 200 iterations, 20 runs, and 50 agents, the HSCA obtained a maximum distance of 849.61 units and a minimum distance of 701.16 units (its optimal result); thus, the optimal result of the HSCA was superior to those of the SCA, PSO, DA, and ALO. In addition, the optimal results of the SCA, PSO, ALO, and DA exhibited deviations of 14.7%, 16.9%, 34.7%, and 37.5%, respectively, from the optimal result of the HSCA; thus, the results of the SCA and PSO were stable and more accurate than those of ALO and DA. In summary,

Table 4: Matrix of travel distances and customer demands for case study 2.

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Demand
1	0	67	54	33	15	50	51	46	40	12	48	61	25	52	42	43	45	57	54	36	29	53	34	44	34	37	4	58	46	16	43	0
2	67	0	121	35	64	116	96	74	48	58	50	85	91	70	31	110	55	10	43	35	48	27	64	71	86	94	69	85	110	82	87	17
3	54	121	0	87	60	17	61	77	88	64	93	86	31	86	94	13	88	111	101	89	79	105	72	78	52	44	52	82	23	39	62	4
4	33	35	87	0	35	81	65	47	34	23	30	61	57	47	11	76	33	25	31	17	16	22	34	53	59	65	35	59	76	49	63	5
5	15	64	60	35	0	60	66	60	28	22	57	75	35	65	45	52	56	54	61	30	37	56	47	29	25	30	18	72	57	22	31	10
6	50	116	17	81	60	0	45	63	89	58	82	70	26	72	87	11	76	106	90	86	70	97	61	84	59	52	47	66	7	39	70	2
7	51	96	61	65	66	45	0	25	89	49	49	26	42	33	65	49	43	88	58	78	50	71	32	95	80	78	48	22	38	53	91	13
8	46	74	77	47	60	63	25	0	76	38	25	16	50	9	43	64	19	66	34	62	31	48	14	88	80	80	44	13	56	55	89	19
9	40	48	88	34	28	89	89	76	0	40	64	92	63	79	44	80	66	39	65	18	46	52	63	23	42	51	44	89	85	50	40	17
10	12	58	64	23	22	58	49	38	40	0	36	53	34	43	30	53	34	48	42	31	17	41	24	50	45	49	13	50	53	28	53	5
11	48	50	93	30	57	82	49	25	64	36	0	35	62	20	22	80	6	44	9	48	20	23	21	82	81	84	47	35	75	62	88	12
12	61	85	86	61	75	70	26	16	92	53	35	0	62	15	56	73	30	79	42	77	46	58	29	104	94	95	59	4	63	69	104	7
13	25	91	31	57	35	26	42	50	63	34	62	62	0	59	63	19	58	81	70	61	48	74	43	61	40	36	22	58	22	14	52	14
14	52	70	86	47	65	72	33	9	79	43	20	15	59	0	41	73	15	64	28	64	33	44	19	93	86	88	51	14	65	63	96	7
15	42	31	94	11	45	87	65	43	44	30	22	56	63	41	0	82	26	23	20	27	17	11	33	64	70	76	43	55	81	58	75	9
16	43	110	13	76	52	11	49	64	80	53	80	73	19	73	82	0	75	100	88	79	67	93	59	74	49	42	41	69	13	30	60	10
17	45	55	88	33	56	76	43	19	66	34	6	30	58	15	26	75	0	49	15	50	20	29	15	82	79	82	44	29	69	58	87	17
18	57	10	111	25	54	106	88	66	39	48	44	79	81	64	23	100	49	0	39	25	39	22	56	62	76	84	59	78	100	72	78	20
19	54	43	101	31	61	90	58	34	65	42	9	42	70	28	20	88	15	39	0	47	25	17	30	84	86	90	54	43	84	69	92	18
20	36	35	89	17	30	86	78	62	18	31	48	77	61	64	27	79	50	25	47	0	31	35	49	39	51	59	39	75	82	50	53	14
21	29	48	79	16	37	70	50	31	46	17	20	46	48	33	17	67	20	39	25	31	0	26	18	62	61	65	29	44	64	44	68	8
22	53	27	105	22	56	97	71	48	52	41	23	58	74	44	11	93	29	22	17	35	26	0	39	74	81	86	54	58	90	69	85	20
23	34	64	72	34	47	61	32	14	63	24	21	29	43	19	33	59	15	56	30	49	18	39	0	75	68	69	32	27	54	45	77	12
24	44	71	78	53	29	84	95	88	23	50	82	104	61	93	64	74	82	62	84	39	62	74	75	0	27	36	48	101	82	47	20	8
25	34	86	52	59	25	59	80	80	42	45	81	94	40	86	70	49	79	76	86	51	61	81	68	27	0	9	36	91	59	27	12	19
26	37	94	44	65	30	52	78	80	51	49	84	95	36	88	76	42	82	84	90	59	65	86	69	36	9	0	38	91	53	26	19	17
27	4	69	52	35	18	47	48	44	44	13	47	59	22	51	43	41	44	59	54	39	29	54	32	48	36	38	0	55	43	15	46	8
28	58	85	82	59	72	66	22	13	89	50	35	4	58	14	55	69	29	78	43	75	44	58	27	101	91	91	55	0	59	65	101	13
29	46	110	23	76	57	7	38	56	85	53	75	63	22	65	81	13	69	100	84	82	64	90	54	82	59	53	43	59	0	36	70	17
30	16	82	39	49	22	39	53	55	50	28	62	69	14	63	58	30	58	72	69	50	44	69	45	47	27	26	15	65	36	0	38	6
31	43	87	62	63	31	70	91	89	40	53	88	104	52	96	75	60	87	78	92	53	68	85	77	20	12	19	46	101	70	38	0	16

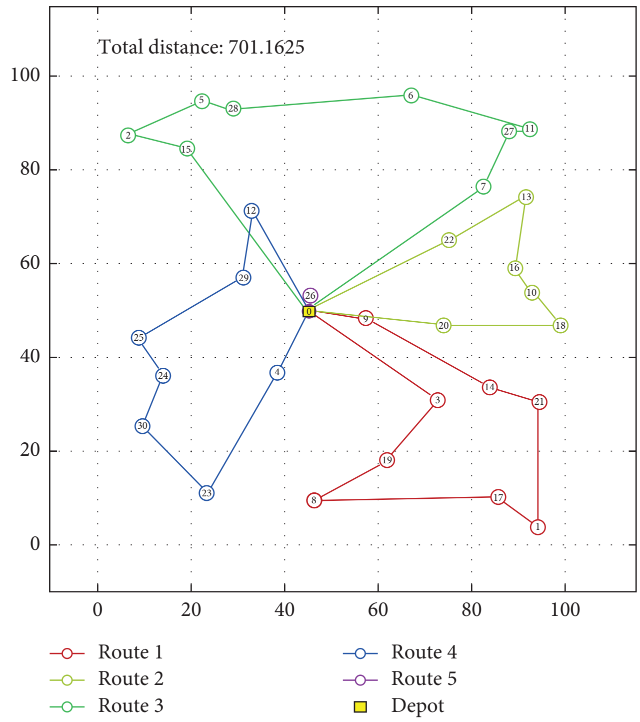

Figure 10: Optimal values obtained by the HSCA after 20 runs in case study 2. Route 1: 0 -> customer3 -> customer19 -> customer8 -> customer17 -> customer1 -> customer21 -> customer14 -> customer 9 -> 0. Route 2: 0 -> customer20 -> customer18 -> customer10 -> customer16 -> customer13 -> customer 22 -> 0. Route 3: 0 -> customer7 -> customer27 -> customer11 -> customer6 -> customer28 -> customer5 -> customer2 -> customer 15 -> 0. Route 4: 0 -> customer12 -> customer29 -> customer25 -> customer24 -> customer30 -> customer23 -> customer 4 -> 0. Route 5: 0 -> customer 26 -> 0.

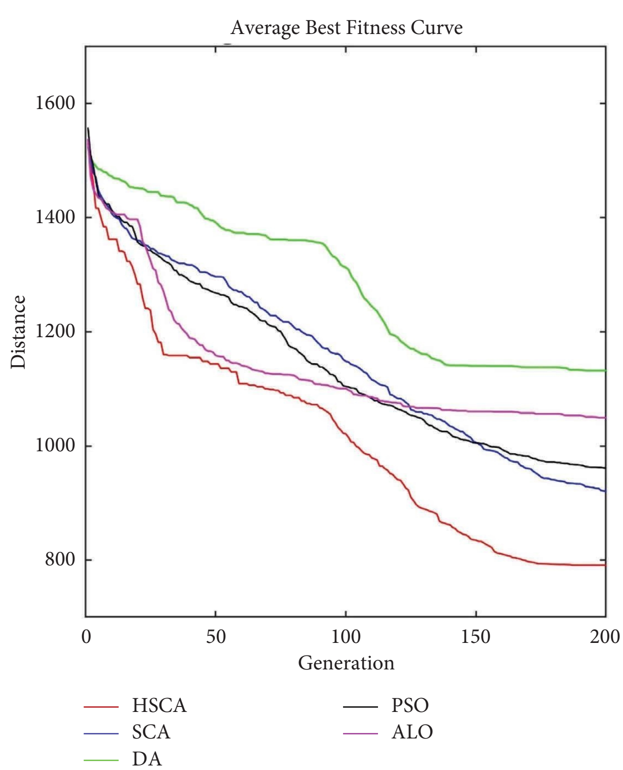

Figure 11: Optimal average values were obtained with different algorithms after 20 runs in case study 2.

Table 5: Best solutions obtained with different algorithms after 20 runs in case study 2.

	Distribution of optimum solutions					Max	Min	Mean	% dev best solution (%)
HSCA	783.81	838.27	748.79	822.01	746.90	**849.61**	**701.16**	**791.59**	**0.0**
	834.89	799.94	849.61	821.61	780.01				
	793.50	701.16	800.64	771.84	821.79				
	806.97	768.76	844.78	719.23	777.37				
SCA	928.49	994.70	1029.05	1026.16	885.60	1119.50	804.05	960.58	14.7
	888.46	1080.39	1029.57	923.55	804.05				
	1119.50	1018.73	946.19	947.45	945.02				
	985.59	959.19	893.72	853.84	952.38				
DA	1122.72	1036.27	1153.79	1095.20	1158.85	1308.59	964.44	1132.01	37.5
	1122.29	1101.74	1079.45	1197.04	1250.21				
	1128.74	1246.84	1061.99	1308.59	969.30				
	964.44	1230.52	1125.79	1138.42	1148.06				
ALO	1165.86	951.44	1049.49	1173.53	971.98	1221.77	944.21	1049.69	34.7
	1066.32	1087.11	947.10	1089.85	1027.15				
	1061.55	1021.48	1061.00	1045.88	944.21				
	1022.59	1007.74	1098.78	1221.77	978.85				
PSO	946.91	987.62	1053.83	1209.33	1042.76	1209.33	819.48	965.35	16.9
	1140.71	953.78	881.51	820.44	924.56				
	983.83	823.34	1068.28	1019.84	978.39				
	1056.59	819.48	872.88	837.93	884.95				

Bold values represent better HSCA search results than other algorithms.

in the problem of cement transport by five vehicles to 30 delivery points, the results obtained by the HSCA were superior to those obtained by the SCA, PSO, ALO, and DA.

5. Conclusion

In this study, an HSCA was developed to solve two case studies of the classical VRP:

(1) In case study 1 [48], the developed HSCA provided stable results and the global optimal solution. Moreover, under the same numbers of search agents and iterations, the HSCA outperformed the DA, PSO, ALO, and SCA.

(2) In case study 2, in which the number of delivery points was 30, the HSCA again provided the global optimal solution and outperformed the aforementioned algorithms. The results of this study indicate that the developed HSCA can be used by cement distributors to optimize their vehicle routing plans to shorten travel distance and reduce travel cost.

In conclusion, a routing planning model for cement transport trucks using HSCA is proposed based on its strong optimization. With case studies from 1 to 2, there will be a more overview of the VRP problem, the increasing complexity of VRP, and comparison of HSCA with the algorithms of previous studies. With the available data and related constraints of the VRP problem, it is found that the study has high practical application in supporting the routing planning for cement transportation vehicles. Further studies will further solve some problems such as delivery with a delivery time window, multiple warehouses, and time-limited delivery of hazardous materials by optimal methods.

Data Availability

The data that support the findings of this study are included within the tables of this paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The researchers would like to express their sincere thanks to Ho Chi Minh City University of Technology-Vietnam National University, Ho Chi Minh City, for their time and material support for this study.

References

[1] K. L. Hoffman, M. Padberg, and G. Rinaldi, "Traveling salesman problem," *Encyclopedia of operations research and management science*, vol. 1, pp. 1573–1578, 2013.

[2] G. Laporte, "What you should know about the vehicle routing problem," *Naval Research Logistics*, vol. 54, no. 8, pp. 811–819, 2007.

[3] S. Anbuudayasankar and K. Mohandas, "Mixed-integer linear programming for vehicle routing problem with simultaneous delivery and pick-up with maximum route-length," *International Journal of Advanced Manufacturing Technology*, vol. 6, no. 1, p. 2, 2008.

[4] C. Qi and L. Hu, "Optimization of vehicle routing problem for emergency cold chain logistics based on minimum loss," *Physical Communication*, vol. 40, Article ID 101085, 2020.

[5] S. Wang, F. Tao, Y. Shi, and H. Wen, "Optimization of vehicle routing problem with time windows for cold chain logistics based on carbon tax," *Sustainability*, vol. 9, no. 5, p. 694, 2017.

[6] W.-Y. Liu, C.-C. Lin, C.-R. Chiu, Y. S. Tsao, and Q. Wang, "Minimizing the carbon footprint for the time-dependentheterogeneous-fleet vehicle routing problem with alternative paths," *Sustainability*, vol. 6, no. 7, pp. 4658–4684, 2014.

[7] K. C. Tan, L. H. Lee, Q. Zhu, and K. Ou, "Heuristic methods for vehicle routing problem with time windows," *Artificial Intelligence in Engineering*, vol. 15, no. 3, pp. 281–295, 2001.

[8] M. Afshar-Bakeshloo, A. Mehrabi, H. Safari, M. Maleki, and F. Jolai, "A green vehicle routing problem with customer satisfaction criteria," *Journal of Industrial Engineering International*, vol. 12, no. 4, pp. 529–544, 2016.

[9] Y. Wang, K. Assogba, J. Fan, M. Xu, Y. Liu, and H. Wang, "Multi-depot green vehicle routing problem with shared transportation resource: integration of time-dependent speed and piecewise penalty cost," *Journal of Cleaner Production*, vol. 232, pp. 12–29, 2019.

[10] W. Zhang, Y. Gajpal, S. S. Appadoo, and Q. Wei, "Multi-depot green vehicle routing problem to minimize carbon emissions," *Sustainability*, vol. 12, no. 8, p. 3500, 2020.

[11] L.-Y. Zhang, M.-L. Tseng, C.-H. Wang, C. Xiao, and T. Fei, "Low-carbon cold chain logistics using ribonucleic acid-ant colony optimization algorithm," *Journal of Cleaner Production*, vol. 233, pp. 169–180, 2019.

[12] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.

[13] R. Zheng, H. Jia, L. Abualigah, Q. Liu, and S. Wang, "Deep ensemble of slime mold algorithm and arithmetic optimization algorithm for global optimization," *Processes*, vol. 9, no. 10, p. 1774, 2021.

[14] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.

[15] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.

[16] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method," *Expert Systems with Applications*, vol. 181, Article ID 115079, 2021.

[17] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *Journal of Bionics Engineering*, vol. 18, no. 3, pp. 674–710, 2021.

[18] I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. Chen, and A. H. Gandomi, "INFO: an efficient optimization algorithm based on weighted mean of vectors," *Expert Systems with Applications*, vol. 195, Article ID 116516, 2022.

[19] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[20] S. Mahdavi, S. Rahnamayan, and K. Deb, "Opposition based learning: a literature review," *Swarm and Evolutionary Computation*, vol. 39, pp. 1–23, 2018.

[21] M. Elsisi, H. G. Zaini, K. Mahmoud, S. Bergies, and S. S. M. Ghoneim, "Improvement of trajectory tracking by robot manipulator based on a new Co-operative optimization algorithm," *Mathematics*, vol. 9, no. 24, p. 3231, 2021.

[22] M. Elsisi and M. A. Ebrahim, "Optimal design of low computational burden model predictive control based on SSDA towards autonomous vehicle under vision dynamics," *International Journal of Intelligent Systems*, vol. 36, no. 11, pp. 6968–6987, 2021.

[23] M. Elsisi, "Improved Grey Wolf Optimizer Based on Opposition and Quasi Learning Approaches for Optimization: Case Study Autonomous Vehicle Including Vision System," *Artificial Intelligence Review*, vol. 55, pp. 5597–5620, 2022.

[24] M. Elsisi, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, "Effective nonlinear model predictive control scheme tuned by improved NN for robotic manipulators," *IEEE Access*, vol. 9, pp. 64278–64290, 2021.

[25] M. Elsisi, M. Soliman, M. Aboelela, and W. Mansour, "GSA-based design of dual proportional integral load frequency controllers for nonlinear hydrothermal power system," *World Academy of Science, Engineering and Technology*, vol. 9, pp. 1142–1148, 2015.

[26] M. Elsisi and M. Q. Tran, "Development of an IoT architecture based on a deep neural network against cyber attacks for automated guided vehicles," *Sensors*, vol. 21, no. 24, p. 8467, 2021.

[27] M.-Q. Tran, M. Elsisi, M. K. Liu et al., "Reliable deep learning and IoT-based monitoring system for secure computer numerical control Machines against cyber-attacks with experimental verification," *IEEE Access*, vol. 10, pp. 23186–23197, 2022.

[28] H. Chen, M. Wang, and X. Zhao, "A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems," *Applied Mathematics and Computation*, vol. 369, Article ID 124872, 2020.

[29] S. K. Dash, S. Mishra, A. Y. Abdelaziz, M. L. Alghaythi, and A. Allehyani, "Optimal allocation of distributed Generators in active distribution networks using a new oppositional hybrid sine cosine muted differential evolution algorithm," *Energies*, vol. 15, no. 6, p. 2267, 2022.

[30] Q. Yang, S.-C. Chu, J.-S. Pan, and C.-M. Chen, "Sine cosine algorithm with multigroup and multistrategy for solving CVRP," *Mathematical Problems in Engineering*, vol. 2020, no. 6, Article ID 8184254, 10 pages, 2022.

[31] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[32] M. Drexl, "Rich vehicle routing in theory and practice," *Logistics Research*, vol. 5, no. 1-2, pp. 47–63, 2012.

[33] M. L. Fisher, A. J. Greenfield, R. Jaikumar, and J. T. Lester, "A computerized vehicle routing application," *Interfaces*, vol. 12, no. 4, pp. 42–52, 1982.

[34] P. Ji, H. Wu, and Y. Wu, "Quadratic programming for the vehicle routing problem," in *Proceedings of the 7th International Symposium on Operations Research and its Applications*, Lijiang, China, November 2008.

[35] C.-C. Wang, Y. Kuo, and P.-Y. Chuang, "Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds," *Computers & Industrial Engineering*, vol. 57, no. 4, pp. 1385–1392, 2009.

[36] M. Reed, A. Yiannakou, and R. Evering, "An ant colony algorithm for the multi-compartment vehicle routing problem," *Applied Soft Computing*, vol. 15, pp. 169–176, 2014.

[37] K. Venkata Narasimha, E. Kivelevitch, B. Sharma, and M. Kumar, "An ant colony optimization technique for solving min–max multi-depot vehicle routing problem," *Swarm and Evolutionary Computation*, vol. 13, pp. 63–73, 2013.

[38] Ç. Koç and I. Karaoglan, "The green vehicle routing problem: a heuristic based exact solution approach," *Applied Soft Computing*, vol. 39, pp. 154–164, 2016.

[39] M. R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, and E. G. Talbi, "A comparative study between dynamic adapted

PSO and VNS for the vehicle routing problem with dynamic requests," *Applied Soft Computing*, vol. 12, no. 4, pp. 1426–1439, 2012.

[40] L. Bodin, B. Golden, A. Assad, and M. Ball, *The State of the Art in the Routing and Scheduling of Vehicles and Crews: Executive Summary*, National Academies of Sciences Engineering and Medicine, Washington, DC, USA, 1981.

[41] Q. Shan and J. Wang, "Solve capacitated vehicle routing problem using hybrid chaotic particle swarm optimization," in *Proceedings of the 2013 Sixth International Symposium on Computational Intelligence and Design*, pp. 422–427, Hangzhou, China, October 2013.

[42] S. Mirjalili, "SCA: a Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[43] H. Nenavath and R. K. Jatoth, "Hybrid SCA–TLBO: a novel optimization algorithm for global optimization and visual tracking," *Neural Computing and Applications*, vol. 31, no. 9, pp. 5497–5526, 2019.

[44] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, pp. 695–701, IEEE, Vienna, Austria, November 2005.

[45] J. V. Roseline and D. Saravanan, "Crossover and mutation strategies applied in job shop scheduling problems," in *Journal of Physics: Conference Series,* IOP Publishing, Bristol, UK, Article ID 12031, 2019.

[46] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[47] N. Singh and S. Singh, "A novel hybrid GWO-SCA approach for optimization problems," *Engineering Science and Technology, an International Journal*, vol. 20, no. 6, pp. 1586–1601, 2017.

[48] W. Zhengchu, L. Jun, Z. Muxun, and F. Jian, "Research in capacitated vehicle routing problem based on modified hybrid particle swarm optimization," in *Proceedings of the 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, pp. 289–293, Shanghai, China, November 2009.