An artificial fish swarm algorithm for solving a bi-objective capacitated vehicle routing problem

Jinling Li[1, 2], Haixiang Guo[1, 3*], Yan Chen[4], Deyun Wang[1], Kejun Zhu[1]

[1]*School of Economics and management, China University of Geosciences, Wuhan, 430074, P.R. China*

[2]*Jiangcheng College, China University of Geosciences, Wuhan, 430200, P.R. China*

[3]*Key Laboratory of Tectonics and Petroleum Resources, China University of Geosciences, Ministry of Education, Wuhan Hubei, 430074, P.R. China*

[4]*School of Distance and Continuing Education, China University of Geosciences, Wuhan, 430074, P.R. China*

Received 11 April 2014, www.tsi.lv

Abstract

The paper focuses on a capacitated vehicle routing problem with two objectives: one is attainment of specific load factor and the other is minimization of total travel cost. Our approach is based on artificial fish swarm algorithm, a swarm-based heuristic, which mimics the foraging behaviour of a fish swarm. After initializing a school of artificial fish, whose validity is guaranteed by a designed repair operator, global optimal solution search is processed through random behaviour, prey behaviour, swarm behaviour, and follow behaviour. Experimental results for a practical distribution instance are reported and show that the artificial fish swarm algorithm performs better than sweep algorithm and genetic algorithm. This paper contributes to the solution methods of vehicle routing problem.

Keywords: Vehicle routing problem, Artificial fish swarm algorithm, Sweep algorithm, Genetic algorithm

1 Introduction

In this paper, we present an artificial fish swarm algorithm for solving a bi-objective capacitated vehicle routing problem with attainment of specific load factor as the first objective and minimization of total travel cost as the second objective. The capacitated vehicle routing problem (CVRP) [1] is a static and basic version of the vehicle routing problem (VRP) [2]. Its objective is to find optimal routes for a fleet of m identical vehicles serving a set of n customers from a single depot. Each vehicle has a maximum capacity $Q_i (i=1,\dots,m)$. The demands q_j of the customers $j=1,2,\dots,n$ are also deterministic and known in advance, and no split deliveries are available. A solution of the CVRP is described as a set of routs, each starting and ending at the depot and satisfying the conditions that each customer is visited only once and the accumulative demand of the customers in a same route for vehicle i limits to the capacity Q_i . A nonnegative cost c_{ij} originally based on the distance d_{ij} exists between a pair of customers (i, j), contributing to the total travel cost which should be minimized. As an extension of the well-known traveling salesman problem (TSP), the CVRP is NP-hard so that only small-sized instances can be solved to optimality using exact algorithms [1, 3]. Thus, considerable problem-specific heuristics and meta-heuristic algorithms including the sweep algorithm [4], the genetic algorithm [5-6], the tabu search [7], the artificial bee colony algorithm [8] and the ant colony algorithm [9] are introduced into the solution methods. However, to our knowledge, the artificial fish swarm algorithm [10] (AFSA) generally adopted for solving continuous problems is scarcely applied to CVRP. In this paper, we endeavor to expand the solution methods of CVRP by adopting AFSA, whose general procedure is illustrated in Figure 1. In real-life applications, the minimization of total distribution cost is often not the only objective. Various other aspects impact the quality of a solution [11]. Our work is just motivated by that kind of appeal from a company named Zhengzhou Coal and Electricity Materials Supply and Marketing Company (ZCEMS&M) in Henan province of China. The company devotes to convey dangerous goods from the depot to 14 coal mines. Without computation, the manager who is charging of distribution dangerous goods in ZCEMS&M used to assign the transport work arbitrarily and the decision making of a route is based on the manager's empirical knowledge. To cutback the enormous operation cost brought by transportation, the manager resorted to our team for a decision support system concerning vehicle routing. One of the constraints they proposed is that the vehicles launch to serve the set of customers only if the total demand exceeds a deterministic load, the percentage of the total demand to the capacity. Therefore, to generalize the problem, we formulate it as a bi-objective CVRP, namely the CVRP

* ***Corresponding author*** e-mail: faterdumk0732@sina.com

with minimum load constraint.

In our work, we use the artificial fish swarm algorithm [10] to solve the considered problem. Finally, we compare the result with that of another two algorithms (sweep algorithm and genetic algorithm) and verify the validity and robustness of AFSA to resolve CVRP.

The paper is organized as follows. Section 2 presents the mathematical model of the problem in this paper. In Section 3, different components of our algorithm is specified and integrated to operate. In Section 4, the algorithm is examined on a practical distribution instance, and the result is compared with that of the other two algorithms in the same circumstance. Section 5, eventually, gives concluding remarks and future research.

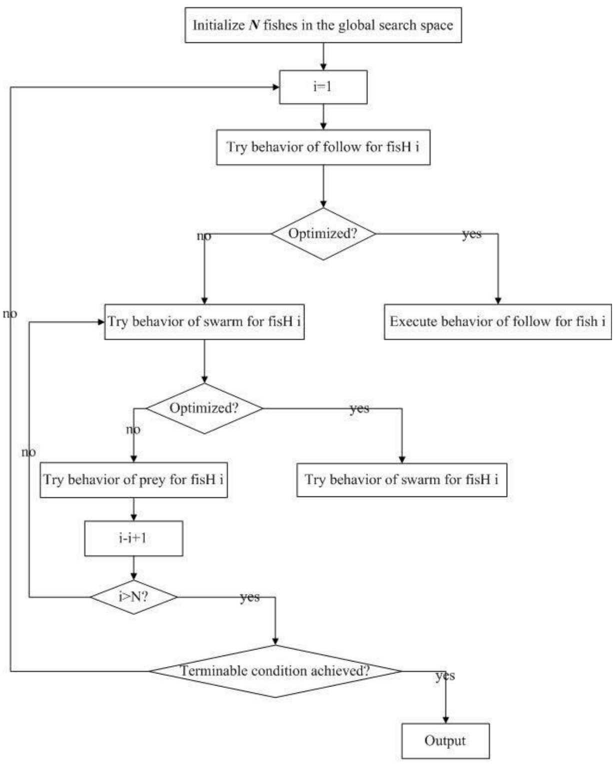

FIGURE 1 General procedure of AFSA

2 Model formulation

This section describes our model for the bi-objective CVRP. It can be seen as a metamorphosis of the model applied by Laporte et al. [12-13] and P. Reiter [14] for the DCVRP and CVRPB. The CVRP with two objectives in this paper (One is attainment of specific load factor and the other is minimization of total travel cost) is formulated as follows. The problem stretches itself on an undirected graph $G=(V,E)$, where $V=\{0,1,\ldots,n\}$ is the set of vertices, namely the depot and customers, and $E=\{\{i,j\}:i,j\in V, i<j\}$ is the set of edges. Index 0 denotes the depot where m available vehicles $M=\{1,\ldots,m\}$ of capacity $Q(v)$ (of which $v\in\{1,\ldots,m\}$ and denotes a vehicle) are located. Meanwhile, the set of customers is given as $V_0=V\setminus\{0\}$. Each customer i has a nonnegative demand $q(i)$. Moreover, to each edge $e\in E$, a cost value c_e, which can also be interpreted as the distribution time or as the length of edge e is associated. The cost matrix C is composed of all the cost values. We assume that C is symmetric and no service time are present. But the elements of C are not supposed to fulfil the triangle inequality (i.e., the distance function of every two customers is not a metric but given as a parameter).

For abbreviation, $\theta(S)$ signifies the set of edges in G with exactly one end vertex in S, e.g., $\theta(S)=\{\{i,j\}\in E: i\in S, j\in V\setminus S\}$, and $\theta(\{i\})$ is simply marked as $\theta(i)$. Furthermore, $\gamma(S)$ denotes the set of edges with both end vertices in S e.g., $\gamma(S)=\{\{i,j\}\in E: i,j\in S\}$. Finally, $(S:\bar{S})$ is the set of edges with one end in S and the other in $\bar{S}$. For each edge e, the decision variable x_e is defined as the multiplicity of edge e being assigned as part of a route, where $x_e\in\{0,1\}$ if e is not incident to depot, otherwise $x_e\in\{0,1,2\}$ (because the situation where some vehicle only serves customer i may exists, and thus edge $\{0,i\}$ occurs twice).

Additionally, ε denotes the specific load factor. For each vehicle v, the decision variable x_v is defined as whether the load of vehicle v exceeds load factor ε, of which $x_v=1$ if it is, otherwise $x_v=0$, and the decision variable x_r is defined as whether the load of vehicle r exceeds 0, of which $x_r=1$ if it is, otherwise $x_r=0$.

The considered bi-objective CVRP of this paper is provided by the following linear bi-objective optimization problem. The problem formulation contains the function $\pi(i,v)$, which is defined as whether vertex i is in the route of vehicle v, and equals 1 if it is, otherwise 0.

$$\min\Big(\sum_{v\in M} Px_v+\sum_{e\in E} c_e x_e\Big), \tag{1}$$

$$s.t. \quad \sum_{e\in\theta(i)} x_e=2 \quad \forall i\in V_0, \tag{2}$$

$$\sum_{e\in\theta(0)} x_e=2\sum_{r\in M} x_r, \tag{3}$$

$$\sum_{i\in V_0} q(i)\pi(i,v)\le Q(v) \quad \forall v\in M, \tag{4}$$

$$\begin{aligned} &x_v\in\{0,1\},\ x_e\in\{0,1\}\ \forall e\notin\theta(0) \\ &x_e\in\{0,1,2\}\ \forall e\in\theta(0) \end{aligned}.$$

Equation (1) defines the two objective functions to be minimized, of which P denotes a constant that is large enough. Equations (2) and (3) convince that exactly two edges are incident to each customer vertex and that accurately $2m$ edges are connected to the depot vertex. Equation (4), finally, restricts that the total demand of the customers that each vehicle serves cannot exceeds its capacity.

It should be noted that with constraint (3), the number of routes is a variable, i.e., when the total quantity of a set of demands is too small and a single vehicle can cover with a load approximating load factor ε, and other vehicles have no need to be launched and their loads equal 0. So, the vehicles available are scheduled dynamically.

3 Algorithmic solution

As a swarm-based heuristic, which mimics the foraging behaviour of a fish swarm, AFSA has outstanding performance in solving complicated practical problems, and becomes eye-catching for its simplicity of simulation. The algorithm searches for the global optimum value through individuals executing various behaviours.

In this section, firstly, separate components of the algorithm are explained, and then how they are woven together to solve the CVRP is described.

3.1 INITIAL SOLUTION

The artificial fish (AF) is the individuals that swim to find the optimal solution. Each AF holds a variable, namely the solution of the problem. Section 3.1.1 presents what is the solution looks like and how to formulate a solution. In Section 3.1.2, we proposed a repair operator to guarantee feasibility of solutions represented by AF.

3.1.1 Solution presentation

Each AF is denoted by a two-dimensional array $x[m][n]$, of which m is the quantity of vehicles and n is the quantity of customers. In this paper, we consider relatively large-scale vehicle routing problem, with more than 2 vehicles. The array can also shortly written as x. Figure 2 illustrates a representation of a CVRP instance with $m=4$ and $n=14$. For each value of the array, $x[i][j](i=1,\ldots,m,\ j=1,\ldots,n)$ denotes the j th customer that the vehicle i serves, e.g., $x[1][2]=4$ and $x[1][5]=0$, because vehicle 1 serves the fourth customer and the fifth customer does not exist.

2	4	7	9	0	0	0	0	0	0	0	0	0	0
8	6	3	11	1	0	0	0	0	0	0	0	0	0
14	5	12	13	10	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIGURE 2 Solution presentation

As Figure 2 shows, vehicle 1 serves customer 2, 4, 7, and 9, and its route can be denoted as 0 – 2 – 4 – 7 – 9 – 0, of which 0 is the depot; vehicle 2 serves customer 8, 6, 3, 11, and 1, and its route can be denoted as 0 – 8 – 6 – 3 – 11 – 1 – 0; vehicle 3 serves customer 14, 5, 12, 13 and 10, and its route can be denoted as 0 – 14 – 5 – 12 – 13 – 10 – 0; vehicle 4 serves none of the customers and is not launched. Obviously, the solution satisfied the constraints of Equation (2) and Equation (3) in our model, see Section 2. To facilitate the following behaviours, we prescribe that all the customers should be located before the 0s, so the customer next to 0 is the last customer the vehicle visits.

To initialize a solution, we assign the 1~ n customers into a stochastic vehicle route. Therein, the customers are randomly ordered to assure the diversity of the fish swarm.

3.1.2 Repair operator

For AF x, when the total demand g of the customers in vehicle i exceeds the capacity $Q(i)$, we need to keep removing the customers backward from the last one until $g<Q(i)$, and record them into the single array $b[n]$. The locations that the removed customers used to sit are filled by 0s. As to the customers stored in array b, we reassign them to the other vehicles and the launched vehicles are at the first consideration.

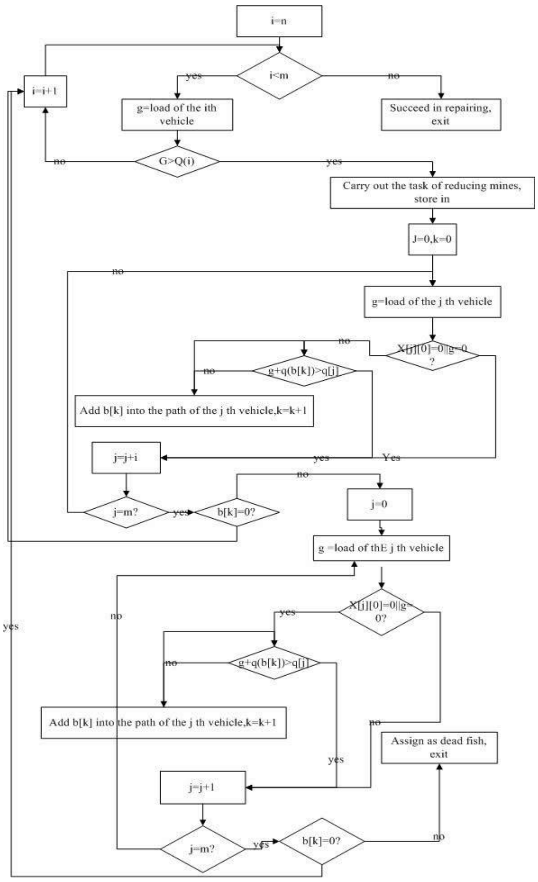

FIGURE 3 Procedure of repairing AF

Notes: $m=4, n=14, visual=n=14,$

$$x=\begin{matrix} 2 & 4 & 7 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 6 & 3 & 11 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 14 & 5 & 12 & 13 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

However, it may happened that when all the vehicle are saturated, some customers are still not reassigned and are stored in array b. In this case, we deem AF x a dead fish. To isolate that dead fish, we put 1~n customers into every row of the array x, so that it is permanently inferior to any live fish and cannot be perceived by the live ones. The concept of perception is explained in Section 3.3.1. Figure 3 illustrates the process of repair operator.

3.2 FOOD CONCENTRATION

We define food concentration as the objective functional values of AF. For abbreviation, $f(x)$ denotes the food concentration of AF x. Equation (5) represents the calculation process of $f(x)$, and the values of the decision variables on the right can be obtained according to the solution x. For detail of the right part, see the mathematical model in Section 2.

$$f(x)=\sum_{v\in M} Px_v+\sum_{e\in E} c_e x_e\ . \tag{5}$$

3.3 BEHAVIOURS OF ARTIFICIAL FISH

The process of AF's optimization is achieved by three kinds of basic behaviours. This section discusses how to realize its perception and behaviours of prey, swarm, and follow.

3.3.1 Perception

Perception ensures that AF is able to perceive the state of other fishes and the surrounding environment. The scope of perception is limited, and *visual* is the threshold. If the distance between AF x and xv, $d(x,xv)$, is bigger than *visual*, AF x cannot perceive xv. As to the distance d, we define it as the degree of similarity between two variables.

$$e_{ij}=\begin{cases}1\ (xv_{ij}-x_{ij}\neq 0)\\ 0\ (xv_{ij}-x_{ij}=0)\end{cases}, \tag{6}$$

$$d(x,xv)=\sum_{i=0}^{M}\sum_{j=0}^{N} e_{ij}\ . \tag{7}$$

Equation (6) and (7) denotes how to calculate the distance of AF xv to focal AF x. As we have designed them as arrays, each AF consists of $m\times n$ values located in their respective rows and columns. If the two values, x_{ij} and xv_{ij}, both of which are in row i and column j, are not equal, e_{ij} equals 1 and the 1 adds to $d(x,xv)$, otherwise e_{ij} equals 0 and the 0 adds to $d(x,xv)$.

It can be seen from the definition of variable that each value of the variable is just a symbol, and does not share digital meaning, thus, the addition and subtraction of variables is meaningless to the distance calculation. Therefore, the general methods calculating the distance of multi-dimensional vector is not appropriate, and may distort the visual perception of AFs. However, the problem-specific method introduced in this paper latch on the essence of distance and is much more logical and the bigger the value of distance, the higher level of heterogeneity between the two variables. When the value is bigger than our given threshold *visual*, the two fish are invisible to each other. The worst condition of the distance calculation occurs when the two distribution schemes are totally dissimilar and the relative distance is $2n$ (The location of customers 1~n are all different, then $d=n+n=2n$). Therefore, *visual* should be smaller than $2n$.

As values of a dead fish do not equal 0 in every location, its distance to any live fish should be no less than $m\times n-n$, and it is invisible for any live one. Meanwhile, behaviours of the live fish are perfectly isolated from the dead.

3.3.2. Behaviour of prey

We decompose the behaviour of prey into three procedures. Firstly, the focal AF x stochastically searches a point xv that is perceivable, legitimate, and feasible. As to guarantee perceivably, the number of customers that each vehicle serves and the route along which each vehicle travels of xv should be similar to that of x at a specific level so that $d(x,xv)\leq visual$. As Section 3.3.1 explained, during the comparison of each value in row i and column j of xv and x, a 1 is added to $d(x,xv)$ in one of the three conditions: (a) $x_{ij}\neq 0$ and $x_{ij}=0$; (b) $xv_{ij}=0$ and $x_{ij}\neq 0$; (c) $xv_{ij}\neq 0$ and $x_{ij}\neq 0$, but $xv_{ij}\neq x_{ij}$. To simplify the illustration, we use s to signify the number of locations in condition (a), s' in condition (b), and $\bar{t}$ in condition (c). In addition, t denotes the number of locations in the condition that $xv_{ij}\neq 0$, $x_{ij}\neq 0$, and $xv_{ij}=x_{ij}$. The distance then can be described by function $d(x,xv)=s+s'+\bar{t}$ and we can derive from the principle of the solution in Section 3.1.1 that $s+t+\bar{t}=n$ and $s'+t+\bar{t}=n$. Therefore, $s=s'$, $d(x,xv)=2s+\bar{t}\leq visual$, and $0\leq s\leq visual/2$, i.e. there should be at least $\bar{s}_{\min}=n-visual/2$ locations where $xv_{ij}\neq 0$ and $x_{ij}\neq 0$. To formulate a xv, firstly we should specify the locations in xv where we allocate the n customers by sequentially selecting $\bar{s}_{\min}$ locations in xv where $x_{ij} \neq 0$ from the first column and randomly distributing the rest $(n-\bar{s}_{\min})$ locations in back of the selected locations. For instance, see Part I in Figure 4, where we present an instance with *visual* equalling n and x equalling the solution presented in Figure 2.

Having fixed the locations, we can get the exact value of s, and thereby the interval of t should be $\left[\max(n+s-visual,0), n-s\right]$. Secondly, we should specify the route of each vehicle, i.e. the values of each row in array xv, by randomly choosing $t_{\min} = \max(n+s-visual,0)$ locations where $xv_{ij} \neq 0$ and $x_{ij} \neq 0$ and assigning xv_{ij} as x_{ij}, and sequentially inserting the rest $(n-t)$ customers that are stochastically ordered into the other blank locations. For instance, see Part II in Figure 4. For feasibility, the repair operator designed in Section 3.1.2 is introduced to remedy xv. Although $d(x,xv)$ might slightly exceed *visual*, we regard the excess acceptable to avoid the further sophisticated and nonessential repair operation.

Part I Location specifying

(1) Sequentially select $\bar{S}_{\min} = n - visual/2 = 7$ location in xv

$$xv = \begin{array}{cccccccccccccc} \_ & \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

(2) Randomly distribute the rest $(n-\bar{s}_{\min}) = 7$ locations in back of the selected locations

$$xv = \begin{array}{cccccccccccccc} \_ & \_ & \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \_ & \_ & \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \_ & \_ & \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Part II Rout specifying

(1) Randomly choose $t_{\min} = \max(n+s-visual,0) = 4$ locations where $xv_{ij} \neq 0$ and $x_{ij} \neq 0$ and assign xv_{ij} as x_{ij}

$$xv = \begin{array}{cccccccccccccc} \_ & 4 & \_ & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \_ & 6 & \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 14 & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \_ & \_ & \_ & \_ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

(2) Arrange the rest $(n-t) = 10$ customers in stochastic order as

$$C_r = \{1,2,11,3,12,5,7,8,13,10\}$$

(3) Sequentially inserts the customers in C_r into the blank locations

$$xv = \begin{array}{cccccccccccccc} 1 & 4 & 2 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 11 & 6 & 3 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 14 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 8 & 13 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\therefore d(x,xv) = 12 < visual.$$

FIGURE 4 Instance of searching a perceivable point

Secondly, xv is estimated whether it is better than x. If it is, the values of variable xv will be recorded in another variable xn signifying the next status of x and originally equal the values of x. Otherwise, AF x repeats the first procedure until a better xv is obtained for try_number times at most. If after repeating try_number times a better xv still cannot be found, AF x will try the random behaviour, which means to exchange the locations of two customers, i and j, stochastically. Customer i and customer j can be in the same route of a vehicle or not, and the updated status will be recorded in xn. The random behavior prevents AFs from trapping into the local optimum and proceeds to the global optimum.

Thirdly, the food concentration of x after trying the behaviour of prey, namely $f(xn)$, is obtained. What to keep an eye on is that trying behaviours does not mean to update x with the optimized status but record the status in variable xn.

In the general artificial fish swarm algorithm applied to solve continuous problems, parameter *step* is set to standardize the behaviours of AF. However, in this CVRP problem, every variable denotes a feasible distribution scheme, so the solution set is not continuous, but discrete, and the values of each variable are not numeric, but nominal. Obviously, the definition of *step* is nonessential to this problem. Setting *step*, and enforcing AFs move in terms of *step*, would illegitimate the variables so that they need significantly complicated repair. Instead of step by step, we enable AFs to jump to better status directly.

3.3.3 Behaviour of swarm

The behaviour of swarm is condensed into four procedures. Firstly, the focal AF x finds n_f friends, all the other AFs perceivable, and records the variables into three-dimensional array $fri\left[fish\_number\right][][]$, of which n_f denotes the quantity of AFs in the perception scope of x and $fish\_number$ denotes the quantity of all AFs.

Secondly, we calculate the centre of these friends who is recorded as a temporary AF xc. For each location, i.e. row i and column j, we calculate the frequencies of 0 and each customer, ranging from 1 to n, and record the frequencies into single-dimensional array $\max[n+1]$. The value of $\max[r]$ signifies the frequency of customer r occurring in this location, and $\max[0]$ the frequency of 0. The customer or 0 who owns the biggest frequency is recorded in xc in the designated location, row i and column j. And the values of xc in other locations are calculated in the same way. Then, two illegal situations may happen to xc, where (a) some customers emerge repeatedly in various locations and (b) some customers never occur. Therefore, we propose a series of remedies.

To remedy situation (a), we record the frequencies of occurrence for $1 \sim n$ customers in locations of xc into single-dimensional array $b[n+1]$, of which the value of $b[i]$ denotes the frequency of customer i in xc, and $b[0]$ represents the frequency of 0; simultaneously, the locations in which a customer occurs again are assigned as 0, recording their row labels and column labels into the two-dimensional array $rc[n][2]$, of which $rc[][0]$ is the row label and $rc[][1]$ is the column label. After that, locations of xc with repeated customer are substituted for 0, and we can locate the places by reading data in array $rc[n][2]$.

To regulate situation (b), we scan values in array b and put customer i, of which $b[i]$ equals 0, in the locations stored in array rc. If some customers in the same boat remain, they are added to the launched vehicle arbitrarily. However, if the locations stored in array rc remain and happen to be in the middle of the route, xc violate the definition laws. A further remedy is necessary. Therefore, we move the customers after 0 forward, and the 0s before customer backward. Moreover, in order to cover the validity of xc, it is corrected by repair operator designed in Section 3.1.2.

Thirdly, we figure out whether (a) xc is superior to x and (b) it is not crowded around xc. If the condition above is satisfied, the values of xc are recorded in xn, otherwise xn remains the same to x. Condition (a) is based on comparing the food concentration of x and xc, namely $f(x)$ and $f(xc)$. Equation (8) denotes condition (b), i.e., if the food concentration of xc multiplied by the quantity of friends of x, n_f, is bigger than the saturation factor δ multiplied by the food concentration of x, AF x thinks that it is too crowded around xc. Equation (9) denotes that the saturation factor δ, one of the parameters of the algorithm, is determined by $n_{\max}$, the expectation of the maximum quantity of individuals we want to see near the extremum, and a, at what level we want the function value of individuals to approximate the extremum. For example, when we have determined that there should be at most 15 AFs around the target point whose food concentrations approximate that of the point at level of 80%, the saturation δ equals 12.

$$f(xc)n_f > \delta f(x), \tag{8}$$

$$\delta = an_{\max},$$

$$0 < a < 1. \tag{9}$$

Fourthly, the food concentration of xn, $f(xn)$, is returned, same to the last procedure of the behaviour of prey.

3.3.4 Behaviour of follow

As to the behaviour of follow, focal AF x firstly picks out the optimal one from its friends whose distance to x is within the perception scope, and we record values of the optimal friend and the quantity of the friends of x into a temporary AF xm and a numerical variable n_f respectively. Then, if xm is superior to x and it is not crowded around xm, namely $f(xm) < f(x)$ and $f(xm)n_f > \delta f(x)$, xm is duplicated by xn, otherwise xn remains the same to x. Finally, the behaviour of follow returns the food concentration of xn like the behaviours above.

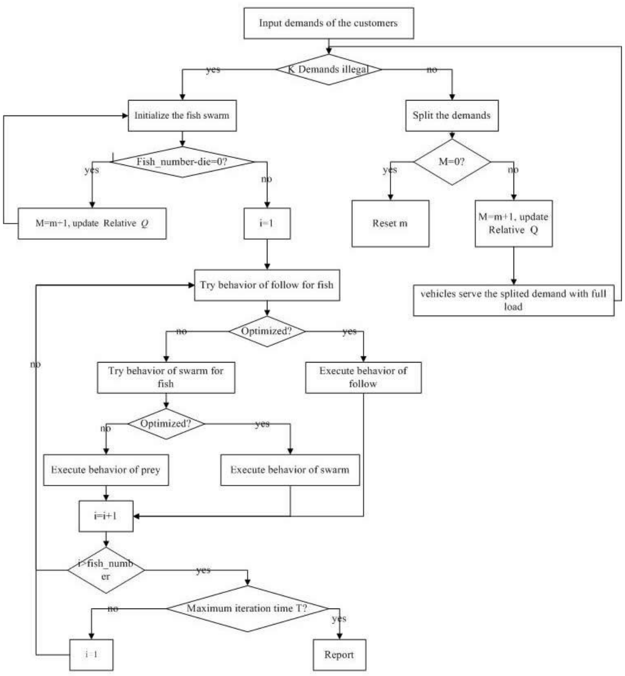

FIGURE 5 Complete flow of AFSA

3.4 EVALUATION OF THE BEHAVIORS

The evaluation of behaviours illustrates the logic of behaviour execution. For AF x, it is commended to try the behaviour of follow fist. If the food consistence returned by this behaviour is smaller, x exits the function directly with the value of $f(xn)$ returned, otherwise tries the behaviour of swarm, and if the food consistence returned is smaller, x also exits the function directly with the value of $f(xn)$ returned, otherwise executes the behaviour of prey, and then exits with the value of $f(xn)$ returned. Variable xn is the next status of AF x, just as the sections of three behaviours clarified.

3.5 ACTION

Action means to update the focal AF x with xn. Although AF x tried many behaviours to optimize itself beforehand, its status has never been updated till now.

3.6 BULLETIN

Bulletin always holds the best record, variable $best\_x$, throughout the algorithm. It is set to compare the variable x put into the bulletin function with $best\_x$, based on their food concentrations. If x is superior to $best\_x$, all the values of x will be duplicated to $best\_x$ to guarantee the optimality of $best\_x$.

3.7 INTEGRATION OF THE COMPONENTS

In this section, we illustrate how to conduct the entire process of the optimization. The algorithm this paper introduced progresses in four procedures. Firstly, we assign all the parameters, since they are the basic standard of the algorithm-processing environment and different sets of parameters lead to significantly distinct efficiency. The parameters are the number of vehicles and customers, m and n, the capacity of each vehicle, $Q(i)$, and the cost matrix, C. Figure 5 illustrates the entire flow of algorithm.

Procedure 1: Input of the demands. The deterministic set of demands activates the algorithm. However, in case the demands surpass the total load ability of vehicles, we should, firstly, judge the legality of the demands. In the algorithm, we tolerate a conditional extent of the demands exceeding the capacity. If a demand is bigger than the capacity, we split it by the capacity and cutback one of the vehicles, so the available vehicles is decreased to $m-1$. The surplus formulates a new demand used to update the demand of the original customer. With the decrease of available vehicles, m may equal 0, then a new circle of fleet scheduling begins and m is reset to the initial one. Then, go to Procedure 2.

Procedure 2: Initialization. For the best record $best\_x$, we set it to the status of a dead fish. Then, the $fish\_number$ fishes in the swarm are initialized, during which all the fishes are compared with $best\_x$ in the process of Bulletin to update the best record. As to the dead fish formulated in period of initialization, we denote the multiplicity of them as die. Therefore, the number of live fish is $fish\_number-die$. If it equals 0, we have to add the vehicles that are available, otherwise go to Procedure 3. For that, we choose to circularly use the current vehicles, meaning the reused vehicle will serve twice.

Procedure 3: Optimization. The artificial fishes are in turns to action the behaviours based on the logic interpreted in evaluation of behaviours section, and update the bulletin instantly. While one cycle of that finishes, one turn of iteration accomplishes. When it adds to the maximum T times, the algorithm terminates.

Procedure 4: Reporting. Variable $best\_x$ holds the optimal solution we find. To make it more comprehensive, we translate $best\_x$ to distribution scheme that can be generally understood by the people outside this work.

4 Computational experiments

We have tested the AFSA algorithm on a practical instance of Zhengzhou Coal and Electricity Materials Supply and Marketing Company (ZCEMS&M). The experiment was performed on 1.86 Gigahertz computer, and the algorithm was coded in Visual C++ 6.0. The number of artificial fishes $fish\_number$, the bound of perception $visual$, the times of independently searching try_number, the maximum iteration time T and the saturation factor δ are set as Table 1.

TABLE 1 Parameter Presentation

$fish\_number$	$visual$	try_number	T	δ
50	16	20	5000	9

4.1 IMPLEMENTATION DETAILS

The dangerous goods distribution problem of ZCEMS&M can be described as that 4 vehicles that locate at a single depot serve 14 customers, namely the coal mines. That kind of vehicle routing problem takes place in the company every day, so the cost reduction brought by computerized optimization is fairly attractive to the company. The information of vehicles is presented in Table 2.

TABLE 2 Information of vehicles

The number of vehicles	4			
Capacity	2	2	2	2
Label	4545	4537	893	763
Oil consumption per kilometre	11.5	11.5	11	11

However, compared to the mathematical model, the oil consumption per kilometre of each vehicle $oil(i)(i=1,\dots,m)$ is added to the property of vehicle. We set oil price p as 6 yuan per litre. Relatively, the food concentration calculation is changed by multiplying the total distance of each route of vehicle i by $6 \times oil(i)$. And the distances between each two customers are a general knowledge among the managers, shown in Table 3. The data serve to fulfil cost matrix C. The company claims that they are unwilling to launch a vehicle when the work load is under 5/6 of the vehicle capacity.

To verify the validity of the algorithm, we test four sets of demands which are shown in Table 4. Then, one of the sets of demands is chosen to test by other algorithms, such as the sweep algorithm (SA) and the genetic algorithm (GA). All of the results are shown in Section 4.2.

4.2 RESULTS ANALYSIS

Figure 6 shows the results of four sets of demands tested by AFSA. We can see that in each instance the optimal solution is obtained by iterating at most 260 times of set 2. The iterations of set 1, 3 and 4 are 22 times, 18 times, 66 times, respectively.

TABLE 3 Distances between each two customers

Distance (km)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
0	0	10	28	23	65	16	16	42	16	21	85	23	35	70	23
1	10	0	76	72	150	52	52	104	42	62	190	60	90	110	65
2	28	76	0	106	74	48	83	28	88	58	186	106	86	146	106
3	23	72	106	0	180	82	82	134	18	92	220	18	120	110	10
4	65	150	74	180	0	122	157	56	162	132	172	180	120	215	180
5	16	52	48	82	122	0	59	76	64	34	162	82	62	117	82
6	16	52	83	82	157	59	0	111	64	69	197	82	97	74	82
7	42	104	28	134	56	76	111	0	116	106	126	134	114	283	134
8	16	42	88	18	162	64	64	116	0	74	202	18	102	112	18
9	21	62	58	92	132	34	69	106	74	0	172	92	72	127	92
10	85	190	186	220	172	162	197	126	202	172	0	220	110	255	220
11	23	60	106	18	180	82	82	134	18	92	220	0	120	130	10
12	35	90	86	120	120	97	97	114	102	72	120	120	0	155	120
13	70	110	146	110	215	74	74	283	112	127	130	130	155	0	120
14	23	65	106	10	180	82	82	134	18	92	220	10	120	120	0

Notes: 1-14 denote the corresponding fourteen coal mines; 1 denotes Peigou; 2 denotes Daping; 3 denotes Zhanggou; 4 denotes Baiping; 5 denotes Micun; 6 denotes Chaohua; 7 denotes Gaocheng; 8 denotes Lugou; 9 denotes Laojuntang; 10 denotes Jinlong; 11 denotes Zhenxing; 12 denotes Cuimiao; 13 denotes Zhaojiazhai; 14 denotes Sanlimeiye; so do the following 1~14 in other figures and tables.

TABLE 4 Four sets of the demands

sets	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	0	0.8	0	0.9	1.3	1.5	0	0	0.3	0.2	1	0	0.6	0.4
2	0.2	0.8	0.3	0.7	0.5	1	1	1	2	0.1	0.1	0.2	0.2	0.3
3	2	0.2	0.5	0.1	1.3	0.1	1.5	1.8	0	0	0.2	0.3	0.4	0.4
4	0.5	1	1.5	0	0.2	0	0.8	0.2	0	1	0	0.1	0.5	1.2

TABLE 5 Comparison results of the Algorithms

Algorithm	**Solution**	**Total cost**	**Average load**	**launched vehicle**
AFSA 1	0 -> 7 -> 2 -> 1 -> 0; 0 -> 8 -> 3 -> 13 -> 5 -> 0; 0 -> 10 -> 12 -> 14 -> 0.	492.18	116.67%	3
AFSA 2	0 -> 1 -> 8 -> 14 -> 0; 0 -> 3 -> 0; 0 -> 13 -> 10 -> 12 -> 0; 0 -> 7 -> 2 -> 5 -> 0.	412.0	87.5%	4
SA	0 -> 7 -> 2 -> 1 -> 0; 0 -> 3 -> 8 -> 0; 0 -> 14 -> 13 -> 5 -> 0; 0 -> 10 -> 12 -> 0.	452.55	87.5%	4
GA	0 -> 12 -> 5 -> 2 -> 7 -> 0; 0 -> 3 -> 0; 0 -> 14 -> 8 -> 1 -> 13 -> 0; 0 -> 10 -> 0.	465.87	83.75%	4

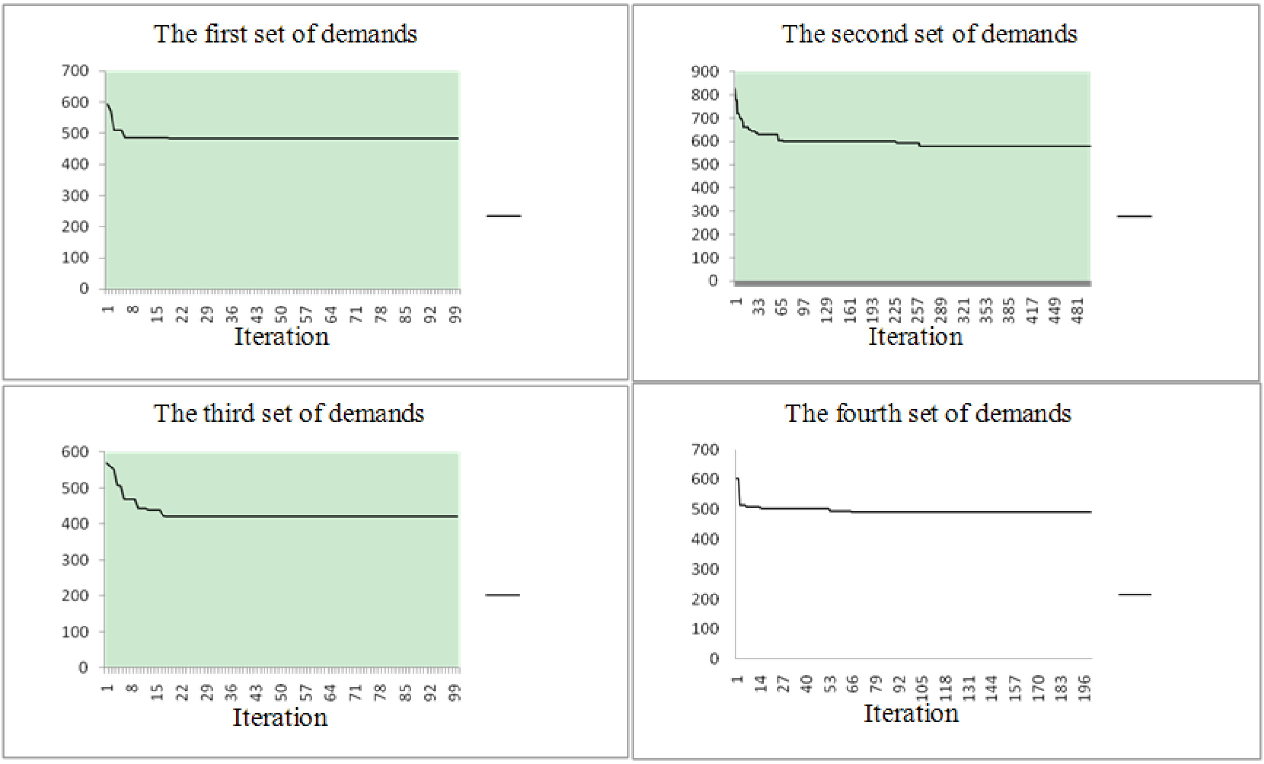

FIGURE 6 Four sets of demand test results of AFSA

The set of demands that we choose to test by other algorithms is the fourth, and Table 5 shows the results. Since the sweep algorithm and genetic algorithm have not consider the first objective of our model, to assure the comparability, we set the load factor of the algorithm as 0 to liberate the minimum load constrain and the artificial fish swarm algorithm in that situation is signed as AFSA 2, with AFSA 1 denotes the original standard. We can see from the table that only AFSA 1 completely satisfied our first objective, but the cost of AFSA 1 is higher than other algorithms. However, if we release the minimum load constrain, AFSA is significantly superior to other two algorithms. Furthermore, it implies that the company may suffer increased transportation cost by achieving the satisfied load.

5 Conclusions

In this paper, we present an artificial fish swarm algorithm (AFSA), a fairly new heuristic, for the capacitated vehicle routing problem (CVRP) with the minimum load constrain. The strategy and optimization process of the AFSA is not complicated and can be applied for practical problem solving appropriately. However, when coming across some general designs of components that violate the well-known laws of real world, the author is suggested to bravely abandon the trivial ones, or innovatively redesign them in terms of the specific problem. For example, we have discarded the component *step* incident with behaviours of the artificial fishes in Section 3.3, and provided a new method of perception process and central point calculation in Section 3.3.1 and 3.3.3 for VRP solving by AFSA. This paper will continuously consider more actual restrictions such as the volume of goods, accidents during the distribution and emergency factors in order to enrich the content of VRP. Meanwhile, more intelligence algorithms can be applied as taboo algorithm, ant colony algorithm, artificial bee colony, etc.

Acknowledgements

This work is supported by the National Natural Science Foundation of China No.71103163, 71103164, 71301153, by Program for New Century Excellent Talents in University, No. NCET-13-1012, by Research Foundation of Humanities and Social Sciences of Ministry of Education of China No.10YJC790071, by the Fundamental Research Founds for National University, China University of Geosciences(Wuhan) No.CUG110 411,CUG120111, G2012002A, CUG140604 by the open foundation for the research centre of resource environment economics in China University of Geosciences (Wuhan) and by the open foundation for Key Laboratory of Tectonics and Petroleum Resources (China University of Geosciences), Ministry of Education No. TPR-2011-11.

References

[1] Toth P, Vigo D (Eds.) 2002 Models, relaxation and exact approaches for the capacitated vehicle routing problem *Discrete Applied Mathematics* **123** 487-512

[2] Dantzig G B, Ramser J H 1959 The truck dispatching problem *Management Science* **10**(6) 80-91

[3] Baldacci R, Toth P, Vigo D 2010 Exact algorithms for routing problems under vehicle capacity constraints *Annals of Operations Research* **175** 213-45

[4] De Bacher B, Furnon V, Shaw P, et al. 2000 Solving vehicle routing problems using constraint programming and metaheuristics *Journal of Heuristics* **6** 501-23

[5] Li Dawei, Wang Li, Wang Mengguang 1999 Genetic algorithm for vehicle routing problem with time windows *Systems Engineering – Theory & Practice* **8** 65-9 *(in Chinese)*

[6] Brysy O, Dullaert W 2002 A fast evolutionary metaheuristic for the vehicle routing problem with time windows *International Journal of Artificial Intelligence Tools* **12**(2) 143-57

[7] Ma Huawei, Yang Shanlin 2008 Improved tabu search algorithm for vehicle routing problem with alternative time windows *Journal of System Simulation* **20**(16) 4454-7

[8] Szeto W Y, Yongzhong Wu, Sin C Ho 2011 An artificial bee colony algorithm for the capacitated vehicle routing problem *European Journal of Operational Research* **215** 126-35

[9] Bell J E, McMullen P R 2004 Ant colony optimization techniques for the vehicle routing problem *Advanced Engineering Informatics* **18** 41-8

[10] Li Xiaolei, Shao Zhijiang, Qian Jixin 2002 An optimizing method based on automous animats: fish-swarm algorithm *Systems Engineering – Theory & Practice* **22**(11) 32-6 (In Chinese)

[11] Jozefowiez N, Semt F, El-Ghazali Talbi 2008 Multi-objective vehicle routing problems *European Journal of Operational Research* **189** 293-309

[12] Laporte G, Desrochers M, Nobert Y 1984 Two exact algorithms for the distance constrained vehicle routing problem *Networks* **14** 161-172

[13] Laporte G, Mercure H, Nobert Y 1986 An exact algorithm for the asymmetrical capacitated vehicle routing problem *Networks* **16** 33-46

[14] Reiter P, Gutjahr W J 2012 Exact hybrid algorithms for solving a bi-objective vehicle routing problem *Central European Journal of Operations Research* **20** 19-43

Authors

Jinling Li, born in April, 1981, Wuhan City, Hu bei Province, P.R. China

Current position, grades: Lecture of Jiangcheng College, China University of Geosciences.
University studies: Management Science and Engineering.
Scientific interest: Study on mathematical model, data mining.
Publications: more than 6 papers published in various journals.

Haixiang Guo, born in September, 1978, Wuhan City, Hu bei Province, P.R. China

Current position, grades: Professor of China University of Geosciences.
University studies: Management Science and Engineering.
Scientific interest: Soft computing.
Publications: more than 30 papers published in various journals.

Yan Chen, born in September, 1976, Wuhan City, Hu bei Province, P.R. China

Current position, grades: the lecturer in China University of Geosciences.
University studies: Knowledge-based Systems and Group decision and Negotiation.
Scientific interest: Decision Making, Comprehensive evaluation and Uncertainty.
Publications: more than 5 papers published in various journals.

Deyun Wang, born in September, 1983, Wuhan City, Hu bei Province, P.R. China

Current position, grades: Associate professor in China University of Geosciences.
University studies: Information system management.
Scientific interest: Schedul optimization.
Publications: more than 10 papers published in various journals.

Kejun Zhu, born in October, 1953, Wuhan City, Hu bei Province, P.R. China

Current position, grades: Professor of China University of Geosciences.
University studies: Mathematics
Scientific interest: Decision Making, Comprehensive evaluation and Uncertainty
Publications: more than 100 papers published in various journals.