18

Krill Herd Algorithm – Modifications and Applications

Ali R. Kashani
Department of Civil Engineering
University of Memphis, Memphis, United States

Charles V. Camp
Department of Civil Engineering
University of Memphis, Memphis, United States

Hamed Tohidi
Department of Civil Engineering
University of Memphis, Memphis, United States

Adam Slowik
Department of Electronics and Computer Science
Koszalin University of Technology, Koszalin, Poland

CONTENTS

18.1 Introduction

The krill herd (KH) algorithm is one of the swarm intelligence algorithms developed by Gandomi and Alavi [1]. KH as a bio-inspired algorithm imitates the herding behavior of krill in the seas using a Lagrangian model and evolutionary operators. Basically, krill individuals tend to be gathered in large swarms for finding food resources. Therefore, herding of the krill addresses two main objectives: (i) increasing krill density, and (ii) reaching food. Considering this fact, the global optimal solution is represented by higher density of krill swarm and shorter distance from the food source. KH algorithm explores the solution space using a swarm of krill as potential solutions. The global optimum solution is the closest krill to the food resource. A krill's movements within the search space is governed by three basic rules: (1) movements induced by other krill; (2) foraging activity; and (3) random diffusion. Moreover, genetic operators i.e., crossover and mutation are utilized in this algorithm to improve the algorithm's performance. The remainder of this chapter is organized accordingly. In Section 18.2 a summary of the fundamentals of KH algorithm is presented. In Section 18.3 some modifications of KH are described. Finally, in Section 18.4 the application of KH algorithm to the optimization of a shallow foundation is discussed.

18.2 Original KH algorithm in brief

In this section the fundamentals of the original KH algorithm are discussed using a step by step pseudo-code.

Algorithm 20 Pseudo-code of the original KH.

determine the $D-th$ dimensional objective function $OF(.)$
determine the range of variability for each $j-th$ dimension $\left[K_{i,j}^{min}, K_{i,j}^{max}\right]$
determine the KH algorithm parameter values such as NK – number of krill, MI – maximum iteration, V_f – foraging speed, D_{max} – maximum diffusion, N_{max} – maximum induced speed
randomly create swarm P which consists of NK krill individuals (each krill individual is a D-dimensional vector)
finding the best krill and its relevant vector
$Iter = 0$
while termination condition not met (here is reaching MI) **do**
    evaluate $X^{food} = \frac{\sum_{i=1}^{NK} \frac{1}{K_i} \cdot X_i}{\sum_{i=1}^{NK} \frac{1}{K_i}}$
    for each i-th krill in swarm P **do**
        $\alpha_i^{target} = 2 \cdot \left(rand + \frac{Iter}{MI}\right) \cdot \widehat{K}_{i,best} \cdot \widehat{X}_{i,best}$

11: $R_{z,i} = \sum_{j=1}^{N} \|X_i - X_j\|$ and $d_{z,i} = \frac{1}{5 \cdot N} \cdot R_{z,i}$
12: **if** $R_{z,i} < d_{z,i}$ and $K(i) \neq K(n)$ **then**
13: $\alpha_i^{local} = \sum_{j=1}^{NN} \frac{K_i - K_j}{K^{worst} - K^{best}} \times \frac{X_j - X_i}{\|X_j - X_i\| + \epsilon}$
14: **end if**
15: $\omega = 0.1 + 0.8 \times \left(1 - \frac{1}{MI}\right)$
16: $N_i^{new} = N^{max} \cdot \left(\alpha_i^{target} + \alpha_i^{local}\right) + \omega \cdot N_i^{old}$
17: $\beta_i^{food} = 2 \cdot \left(1 - \frac{Iter}{MI}\right) \cdot \widehat{K}_{i,food} \cdot \widehat{X}_{i,food}$
18: $\beta_i^{best} = \widehat{K}_{i,best} \cdot \widehat{X}_{i,best}$
19: $F_i^{new} = V_f \cdot \left(\beta_i^{food} + \beta_i^{best}\right) + \omega \cdot F_i^{old}$
20: $D_i = D^{max} \cdot \left(1 - \frac{Iter}{MI}\right) \cdot \delta$
21: $\frac{dX_i}{dt} = N_i^{new} + F_i^{new} + D_i$
22: $X_i = crossover\left(X_i, X_c\right)$
23: $X_i = mutation\left(X_i, X_{best}, M_u\right)$
24: $X_i^{new} = X_i + \frac{dX_i}{dt}$
25: **end for**
26: update best-found solution
27: $Iter = Iter + 1$
28: **end while**
29: post-processing the results

Here, KH algorithm deals with a D-dimensional objective function $OF(.)$ as demonstrated in Algorithm 20. At the second step the permitted j-th variable's domain for i-th krill is defined as $\left[K_{i,j}^{min}, K_{i,j}^{max}\right]$. In step 3, the necessary parameters for KH algorithm are initialized accordingly; NK is number of krill, MI is the maximum iteration, V_f is the foraging speed, D^{max} is the maximum diffusion speed, N_{max} is the maximum induced speed. In the original paper, D^{max} varies between 0.002 and 0.010 (ms^{-1}), V_f and N_{max} are recommended to be 0.02 and 0.01 (ms^{-1}), respectively. In step 4, an initial swarm of NK number of krill is reproduced randomly within the valid solution domain. Each krill is represented by a D-dimensional vector as follows:

$$K_i = \{k_{i,1}, k_{i,2}, ..., k_{i,D}\} \tag{18.1}$$

where $k_{i,1}$ to $k_{i,D}$ are decision variables varying between $K_{i,j}^{min}$ and $K_{i,j}^{max}$.

In step 5, the global best solution and its relevant vector are determined. After that, the current iteration counter is initialized as zero. Now, the main loop of KH algorithm will be started. In step 8, the virtual food location is evaluated. In the original paper [1], the center of food is proposed to be evaluated based on distribution of the krill individuals' fitness. It is somehow like evaluating "*center of mass*". As mentioned previously, the motion of krill individuals within the solution space can be characterized by three basic rules: I. Movement induced by the other krill individuals (N_i); II. Foraging action; and III. Random diffusion.

In KH the following time-dependent Lagrangian model is developed to explore our D-dimensional search space:

$$\frac{dX_i}{dt} = F_i + N_i + D_i \tag{18.2}$$

To address Equation (18.2), step 9 in Algorithm 20 is devoted to illustrating krill individuals' pace through an iterative loop. The first action is to evaluate N_i at first. To this end, in step 10, α_i^{target} results from the best-found solution to define the target direction effect. Another important effect is imposed by the neighbors through local effect α_i^{local}. Before that, the sensing distance for each i-th particle ($d_{s,i}$) is computed based on the distance between this particle and all of the others as shown in step 11. In step 13, α_i^{local} is determined to consider the effect of neighbors. In KH algorithm two different inertia weights are defined; inertia weight for induced motion by other krill individuals (ω_n) and inertia weight of the foraging motion (ω_f). Both of ω_n and ω_f vary in the range of $[0, 1]$, though in this study a time-dependent value is considered for both as shown in step 15. In step 16, the movement resulting from other krill individuals (N_i^{new}) is calculated. To examine foraging motion (step 19), we need two important factors, food attractiveness (β_i^{food}) and the effect of the best fitness of the i-th krill (β_i^{best}). We evaluate β_i^{food} and β_i^{best} in steps 17 and 18, respectively. In step 20, the final movement caused by physical diffusion of the krill individuals is estimated where δ is the random directional vector. The next position of the krill individuals in an interval of $\frac{dX_i}{dt}$ is evaluated in step 21. However, before updating the solutions two evolutionary operators, crossover and mutation, are applied to the current generation in steps 22 and 23, respectively. In step 24, new positions are generated by applying the term of $\frac{dX_i}{dt}$ to the current solutions. The best-found solution is updated in step 26. Finally, the best-found solution is proposed when the termination criteria are satisfied.

18.3 Modifications of the KH algorithm

18.3.1 Chaotic KH

In original KH, tuning inertia weights (ω_n, ω_f) play an important role in the final performance of this algorithm. Higher values of inertia weights push the algorithm toward exploration and lower values encourage the KH algorithm to focus on exploitation. Wang et al. [2] proposed a chaotic KH (CKH) using a chaotic-based pattern for adjusting those parameters to reach faster convergence. In this way, the authors normalized the range of the chaotic map between 0 and 1. CKH updates the values of inertia weights in each iteration using 12 different chaotic maps collected in Table 18.1. Wang et al. [2] applied

TABLE 18.1
Different chaos maps.

ID	Name	Formulation
1.	Chebyshev	$X_{i+1} = cos\left(kcos^{-1}\left(X_i\right)\right)$
2.	Circle	$X_{i+1} = mod\left(X_i + b - \left(\frac{a}{2\pi}\right) sin\left(2\pi X_i\right), 1\right)$
3.	Gaussian	$X_{i+1} = \begin{cases} 0 & X_i = 0 \\ \frac{1}{X_i mod(1)} & \text{otherwise}, \frac{1}{X_i mod(1)} = \frac{1}{X_i} - \left[\frac{1}{X_i}\right] \end{cases}$
4.	Intermittency	$X_{i+1} = \begin{cases} \epsilon + X_i + CX_i^n & 0 < X_i \leq p \\ \frac{X_i - p}{1-p} & p < X_i < 1 \end{cases}$
5.	Iterative	$X_{i+1} = sin\left(\frac{a\pi}{X_i}\right)$
6.	Liebovitch	$X_{i+1} = \begin{cases} \alpha X_i & 0 < X_i \leq p_1 \\ \frac{p - X_i}{p_2 - p_1} & p_1 < X_i \leq p_2 \\ 1 - \beta\left(1 - X_i\right) & p_2 < X_i \leq 1 \end{cases}$
7.	Logistic	$X_{i+1} = \alpha X_i\left(1 - X_i\right)$
8.	Piecewise	$X_{i+1} = \begin{cases} \frac{X_i}{p} & 0 \leq X_i < p \\ \frac{X_i - p}{0.5 - p} & p \leq X_i < 0.5 \\ \frac{1 - p - X_i}{0.5 - p} & 0.5 \leq X_i < 1 - p \\ \frac{1 - X_i}{p} & 1 - p \leq X_i < 1 \end{cases}$
9.	Sine	$X_{i+1} = \frac{a}{4} sin\left(\pi X_i\right)$
10.	Singer	$X_{i+1} = \mu(7.86X_i - 23.31X_i^2 + 28.75X_i^3 - 13.302875X_i^4), \mu = 1.07$
11.	Sinusoidal	$X_{i+1} = aX_i^2 sin\left(\pi X_i\right)$
12.	Tent	$X_{i+1} = \begin{cases} \frac{X_i}{0.7} & X_i < 0.7 \\ \frac{10}{3}\left(1 - X_i\right) & X_i \geq 0.7 \end{cases}$

their proposed algorithm to 14 benchmark functions. It was mentioned in their study that the results recorded considerable superiority of CKH over original KH algorithm.

18.3.2 Levy-flight KH

Metaheuristic optimization algorithms deal with objective function based on two important strategies: exploration and exploitation. Any attempt toward finding an appropriate balance between them improves the algorithm's performance effectively. Wang et al. [3] proposed levy-flight krill herd (LKH) using local levy-flight operator to strengthen the exploitation ability of original KH. Wang et al. [3] utilized the following random walking steps:

$$dX_i^{t+1} = X_i^{t+1} + \beta L\left(s, \lambda\right) \tag{18.3}$$

$$L\left(s, \lambda\right) = \frac{\lambda \Gamma\left(\lambda\right) sin\left(\frac{\pi \lambda}{2}\right)}{\pi} \times \frac{1}{s^{1+\lambda}}, \left(s, s_0 > 0\right)$$

where β is the step size scaling factor and should have a positive value. In LKH, a local Levy flight strengthens the exploitation ability of the KH algorithm by providing much longer step length for searching around the best-found solution.

In the proposed local Levy flight, a time-dependent step size is proposed by the following equation:

$$\alpha = \frac{A}{t^2} \tag{18.4}$$

where A is the maximum step size and $Iter$ is the current iteration number.

For updating the new position of the krill, a vector of random numbers ($random$) with the size of problem dimension (D) will be produced and each d-th dimension updated as follows:

$$X_i^{t+1} = \begin{cases} \alpha \times \frac{dX_i}{dt} + X_{D-r+1} & random\,(d) \leq 0.5 \\ \alpha \times \frac{dX_i}{dt} - X_{D-r+1} & random\,(d) > 0.5 \end{cases} \tag{18.5}$$

where r is a random number. The current solution would be replaced by V_i in case of improvement.

This LKH was tested through 14 benchmark functions and the results are compared to several optimization algorithms. The authors claimed that their proposed modified algorithm achieved better performance rather than the original KH algorithm.

18.3.3 Multi-stage KH

In another effort, Wang et al. [4] tried to boost the KH algorithm by balancing between exploration and exploitation. Original KH proposed a satisfying performance during the search of solution space (exploration). Therefore, any attempt toward strengthening the exploitation ability of KH can be helpful to increase its performance. Wang et al. [4] proposed the multi-stage KH (MSKH) by mounting the original KH with a local mutation and crossover (LMC) operator to intensify the local search concentration. In this way the MSKH algorithm searches the solution space based on different stages. In the first stage, the original KH shrinks the search space as much as possible and in the second stage LMC operator is enlisted to search around the limited space resulting from the first stage more accurately. To be more exact, in the first stage, the position of krill in a herd will be updated using the step size $\frac{dX_i}{dt}$ (line 21 in Algorithm 20). After that, LMC operator modifies the d-th element of each i-th krill's position for all the krill in a herd following the below equation:

$$X_i^{t+1} = \begin{cases} X_{best}\,(d) & random\,(d) \leq 0.5 \\ X_{best}\,(r) & random\,(d) > 0.5 \end{cases} \tag{18.6}$$

where X_i^{t+1} is the current position of the i-th krill, $X_{best}(d)$ is the d-th element of the best solution and $X_{best}(r)$ is the r-th element of the best solution. To validate the proposed algorithm, MSKH is tackled for 25 benchmark functions and compared to several optimization algorithms. As the authors in [4] mentioned, their proposed algorithm, MSKH, performed very well especially in dealing with complex multimodal problems.

18.3.4 Stud KH

To compensate for the weakness of original KH in exploitation, Wang et al. [5] proposed the stud krill herd (SKH) algorithm. In fact, SKH by imitating the stud genetic algorithm tends to use the best-found solution for crossover in each iteration. To be more exact, Wang et al. [5] by incorporating the stud selection and crossover (SSC) operator to the KH algorithm improved the ability of this algorithm to evade local minima. Based on this methodology, for updating the positions of all the krill two different attitudes may be selected. First, the step size defined in the original KH, $\frac{dX_i}{dt}$, will be evaluated. After that, by applying the crossover operator the best krill modifies one of the selected solutions by the selection operator. Next, the quality of the reproduced offspring will be evaluated and in case of improvement the krill position will be updated. Otherwise, the krill individual will move to the next position by adding the term of step size to the current position. This algorithm was tested using 22 benchmark functions and based on the authors' assertion; this simple modification resulted in better performance of the algorithm and increasing accuracy of the global optimality.

18.3.5 KH with linear decreasing step

Gandomi and Alavi [1], developed the basic KH method and they defined $\frac{X_i}{dt}$ which works as a scale factor of the speed vector and it depends on search space and a constant parameter C_t. Li et al. [6] tried to improve the basic method of Gandomi and Alavi [1] by analyzing the parameter C_t. They have conducted various experiments on KH method to investigate the effect of C_t size on search space. They figured that the C_t is large enough that the KH searches entire space, but it cannot search local space accurately. On the other hand when the C_t is small the step will be smaller and the KH can search local space, but it misses the global area. Li et al. [6] claimed that if the step size of C_t can be controlled by increasing the number of iterations, KH can search both the local and global area carefully. By increasing iterations in number, the C_t becomes linearly smaller in each iteration until it covers both the local and global area. The proposed new C_t is provided in Equation (18.7):

$$Ct(t) = Ct_{max} - \frac{Ct_{max} - Ct_{min}}{MI} \cdot t \tag{18.7}$$

where Ct_{max} and Ct_{min} are the maximum and minimum values of Ct, and MI is the maximum number of iterations and t is the current number of iterations.

18.3.6 Biography-based krill herd

Since the basic KH method cannot produce an appropriate convergence all the time, in 2012 Gandomi and Alavi [1] tried to improve the performance of basic KH convergence by adding genetic reproduction mechanisms to the basic KH algorithm. However, sometimes the exploitation ability of KH is still not satisfactory. Wang et al. [7] combined the krill migration (KM) operator with KH to improve the exploitation ability of KH. This method is known as biography-based krill herd (BBKH). Algorithm 21 shows the KM operator steps used in the BBKH.

Algorithm 21 Krill migration (KM) operator.

Select i-th krill (its position X_i) without probability based on k_i
if $rand\,(0,1) < k_i$ **then**
    for $j = 1$ to d (all elements) **do**
        Select X_j with probability based on l_j
        if $rand\,(0,1) < l_j)$ **then** Randomly select an element r from X_j Replace a random element in X_i with r
        end if
    end for
end if

To keep the krill optimal all the time, Wang et al. [7] added a kind of elitism to the algorithm. Therefore, the BBKH algorithm was developed by combining KM and concentrated elitism in the KM algorithm as can be seen in Algorithm 22.

Algorithm 22 Biogeography-based KH method.

Initialization. Set the generation counter $t = 1$; initialize the population P of NP krill randomly; set V_f, D^{max}, N^{max}, S_{max}, and p_{mod}.
Fitness evaluation. Evaluate each krill.
while $t < MI$ **do**
    Sort the krill from best to worst
    Store the best krill
    for $i = 1$ to NP (all krill) **do**
        Perform the three-motion calculation
        Update the krill position by $X_i\,(t + \Delta t) = X_i\,(t) + \Delta t \frac{dX_i}{dt}$
        Fine-tune $X_i + 1$ by performing KM operator in Algorithm 20
        Evaluate each krill by $X_i + 1$
    end for
    Replace the worst krill with the best krill

Sort the krill and find the current best
$t = t + 1$
end while
Output the best solutions

18.4 Application of KH algorithm for optimum design of retaining walls

18.4.1 Problem description

In nature there are many cases in which a soil slope (natural or artificial) is not stable due to the inherent angle of inclination. A retaining wall is one of the structures that are used to stabilize slope. Since this type of structure is used for various applications such as highways, railways, bridges, etc., optimum cost and weight are very important factors that affect the design process of retaining walls. Besides the economical aspect of the design procedure of retaining walls, also a retaining wall must be designed to meet the geotechnical and structural requirements. There are 12 variables (please see Figure 18.1) that should be considered in order to reach an optimal design of a retaining wall; eight of those variables ($X1$ to $X8$) are wall geometry related and are expressed as: width of the base ($X1$), toe width ($X2$), footing thickness ($X3$), thickness at the top of the stem ($X4$), base thickness ($X5$), the distance from the toe to the front of shear key ($X6$), shear key width ($X7$), shear key depth ($X8$). The other four variables ($R1$ to $R4$) are steel; reinforcement dependent in which the vertical steel reinforcement is in the stem ($R1$), the horizontal steel reinforcement of the toe and heel ($R2$ and $R3$, respectively), and the vertical reinforcement of the shear key ($R4$).

Each phase of retaining wall design (geotechnical and structural) is investigated for various failure modes. In the geotechnical phase, the resistance of the wall is evaluated against overturning, sliding, and bearing using the factor of safety. The structural design phase is assessed for shear and moment failure of stem, heel, toe, and shear key. The main step of reaching an optimal design is defining an appropriate objective function. The objective function that is going to be introduced in this chapter, was presented by Saribas and Erbatur [8] for minimizing cost and weight of the wall as provided in Equations (18.8) and (18.9):

$$f_{cost} = C_s W_{st} + C_c V_c \tag{18.8}$$

$$f_{weight} = W_{st} + 100 V_c \gamma_c \tag{18.9}$$

where C_s is the unit cost of steel, C_c is the unit cost of concrete, W_{st} is the weight of reinforcing steel, V_c is the volume of concrete, and γ_c is the unit weight of concrete.

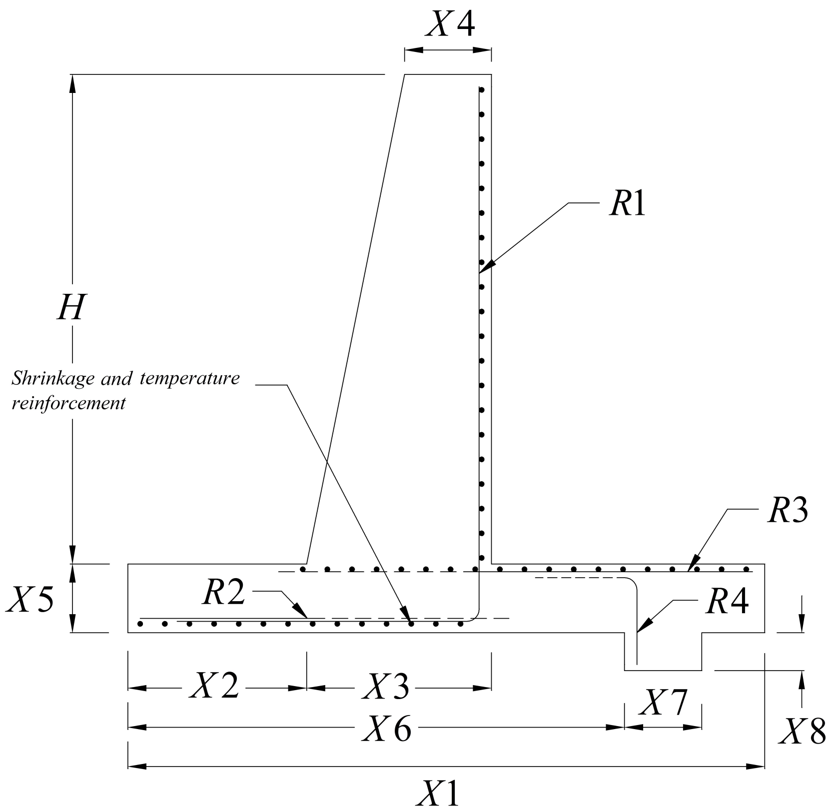

FIGURE 18.1
Reinforced cantilever retaining wall typical geometry and input parameters.

The other predominant factor of each optimization problem is constraints which define satisfying certain conditions to design a safe and stable retaining wall. The constraints can be divided into geometrical, structural, and geotechnical which are included in boundary inequality constraints. According to Saribas and Erbatur [8] and Gandomi et al. [9, 10], the boundary limitations contain the variables that will be satisfied by changing wall geometry. Inequality constraints check the final structural and geotechnical strength and stability. All the structural strength and geotechnical stability limitations are defined as the constraints summarized in Table 18.2.

18.4.2 How can KH algorithm be used for this problem?

In this sub-section we plan to introduce an automatic process for optimum design of retaining walls based on KH algorithm. In the first step a population

TABLE 18.2
Inequality constraints.

Constraint	Function
$g_1(x)$	$\frac{FS_{Odesign}}{FS_O} - 1 \leq 0$
$g_2(x)$	$\frac{FS_{Sdesign}}{FS_S} - 1 \leq 0$
$g_3(x)$	$\frac{FS_{Bdesign}}{FS_B} - 1 \leq 0$
$g_4(x)$	$q_{min} \geq 0$
$g_{[5-8]}(x)$	$\frac{M_u}{M_n} - 1 \leq 0$
$g_{[9-12]}(x)$	$\frac{V_u}{V_n} - 1 \leq 0$
$g_{[13-16]}(x)$	$\frac{A_{smin}}{A_s} - 1 \leq 0$
$g_{[17-20]}(x)$	$\frac{A_s}{A_{smax}} - 1 \leq 0$
$g_{21}(x)$	$\frac{X_2+X_3}{X_1} - 1 \leq 0$
$g_{22}(x)$	$\frac{X_6+X_7}{X_1} - 1 \leq 0$
$g_{23}(x)$	$\frac{l_{dbstem}}{X_5-cover} - 1 \leq 0$ or $\frac{l_{dbstem}}{X_5-cover} - 1 \leq 0$
$g_{24}(x)$	$\frac{l_{dbtoe}}{X_1-X_2-cover} - 1 \leq 0$ or $\frac{12d_{btoe}}{X_5-cover} - 1 \leq 0$
$g_{25}(x)$	$\frac{l_{dbheel}}{X_2+X_3-cover} - 1 \leq 0$ or $\frac{12d_{bheel}}{X_5-cover} - 1 \leq 0$
$g_{26}(x)$	$\frac{l_{dbkey}}{X_5-cover} - 1 \leq 0$ or $\frac{l_{dhkey}}{X_5-cover} - 1 \leq 0$

of krill with the size of NK is initialized. Each krill is a randomly created 12-dimensional vector. Before estimating the objective functions, we need to evaluate several constraints defined by standard codes for the whole swarm. The number of violations will be added to the objective values as a penalty term. Then, we find the best krill individual with the lowest cost value.

Next, the time-dependent additive for updating the position of the krill swarm is evaluated. To this end, we need to evaluate three motion types according to the rules mentioned in Section 18.2: movement caused by other krill (N_i); foraging activity (F_i); random diffusion (D_i). To evaluate N_i, two main factors, α_i^{target} (the effect of the best krill) and α_i^{local} (the effect of neighbors), are required. Foraging motion (F_i) results from the food attractiveness and effect of the best-found solution. Food attractiveness is directly related to the location of the food resources which is proposed to be the center of food. Now, we evaluate D_i to provide random movements of the krill individuals. This randomness follows an annealing-based rule which is decreased as the generation proceeds. Before updating the position of the swarm, we need to apply two genetic operators, crossover and mutation, to the current population. At the final step, we evaluate the penalized objective function and determine the best-found solution. This procedure will be iterated until satisfying the termination criteria.

18.4.3 Description of experiments

In this section our proposed methodology is tested by means of a set of numerical retaining wall models. To evaluate the performance of the presented algorithm, two case studies utilized from the study were done by Saribas and Erbatur [8]. ACI 318-05 [11] requirements are considered in the design for steel reinforcement. To set up the experiments, each algorithm was run 101 times, and population size and number of iterations are inserted as 50 and 1000, respectively. In the first example, the retaining wall was designed twice for both objective functions of cost and weight; also, the sensitivity of designs was investigated for different surcharge load, backfill, slope, and friction angle. Since the first example did not include base shear key, the second example designed a retaining wall for two cases of base shear key, one with base shear key and one without base shear key by optimizing cost and weight. The structural and geotechnical design parameters are summarized in Table 18.3. Moreover, boundaries limitations for the input parameters are gathered in Table 18.4. In both examples the best, the worst, mean, and standard deviation (SD) values were obtained and will be discussed in results section.

18.4.4 Results obtained

In this chapter the optimum design of retaining wall optimization is tackled for further discussion. Due to the stochastic basis of metaheuristic algorithms

TABLE 18.3
Utilized soil parameters for the case study.

Input parameters	Symbol	Value		Unit
		Example 1	Example 2	
Stem height	H	3.0	4.5	m
Reinforcing steel yield strength	f_y	400	400	MPa
Concrete compressive strength	f_c	21	21	MPa
Concrete cover	C_c	7	7	cm
Shrinkage and temperature reinforcement percentage	ρ_{st}	0.002	0.002	-
Surcharge load	q	20	30	kPa
Backfill slope	β	10	0	°
Internal friction angle of retained soil	ϕ	36	36	°
Internal friction angle of base soil	ϕ'	0	34	°
Unit weight of retained soil	γ_s	17.5	17.5	kN/m^3
Unit weight of base soil	γ'_s	18.5	18.5	kN/m^3
Unit weight of concrete	γ_c	23.5	23.5	kN/m^3
Cohesion of base soil	c	125	0	kPa
Depth of soil in front of wall	D	0.5	0.75	m
Cost of steel	C_s	0.4	0.4	\$/kg
Cost of concrete	C_c	40	40	$\$/m^3$
Factor of safety for overturning stability	$SF_{Odesign}$	1.5	1.5	-
Factor of safety for sliding	$SF_{Sdesign}$	1.5	1.5	-
Factor of safety for bearing capacity	$SF_{Bdesign}$	3	3	-

TABLE 18.4
Design variables (DVs) permitted domain.

DVs	Unit	Example 1		Example 2	
		Lower Bound	Upper Bound	Lower Bound	Upper Bound
$X1$	m	1.3090	2.3333	1.96	5.5
$X2$	m	0.4363	0.7777	0.65	1.16
$X3$	m	0.2000	0.3333	0.25	0.5
$X4$	m	0.2000	0.3333	0.25	0.5
$X5$	m	0.2722	0.3333	0.4	0.5
$X6$	m	-	-	1.96	5.5
$X7$	m	-	-	0.2	0.5
$X8$	m	-	-	0.2	0.5
$R1$	-	1	223	1	223
$R2$	-	1	223	1	223
$R3$	-	1	223	1	223
$R4$	-	-	-	1	223

TABLE 18.5
Design cost values for numerical case studies.

KH optimization algorithm	Cost ($/m)			
	Best	Worst	Mean	SD
Example 1	73.17	79.93	74.17	1.12
Example 2 (Case I)	164.62	187.24	170.31	4.15
Example 2 (Case II)	162.46	176.62	165.92	2.57
KH optimization algorithm	Weight (kg/m)			
	Best	Worst	Mean	SD
Example 1	2667.23	2738.08	2684.66	11.83
Example 2 (Case I)	5661.94	6007.72	5719.46	50.54
Example 2 (Case II)	5556.63	5704.27	5603.84	30.67

the algorithm is run 101 times and the results for both examples one and two are reported based on best, worst, mean, standard deviation and median in Table 18.5. The total number of function evaluations is 50,000 for all the cases. In the first case, a 3 m-tall retaining wall design without a base shear key was considered for both low-cost and low-weight design. The results have been kept unchanged after 817-th and 728-th generations for low-cost and low-weight designs, respectively. For the second case study, a 4.5 m-tall retaining wall is considered. Convergence history proved that the results remained unchanged after 872-nd and 958-th generations. In the third numerical simulation, the second case is reconsidered by an additional base shear key. As expected, this additive element causes lower cost and weight design. The final best solution of the KH algorithm in this experienced no improvement after the 932nd and

985th iterations for low-cost and low-weight design, respectively. The convergence history of KH algorithm demonstrated that it reached valid designs which satisfy all the geotechnical and structural limitations since the first iteration for both low-cost and low-weight designs over all the case studies.

18.5 Conclusions

In this chapter, the krill herd algorithm with a step-by-step description was presented in detail. Next, several modifications and improvements of this algorithm were demonstrated. Finally, KH algorithm was applied to one real-world engineering problem. The retaining wall design procedure as one of the important and complicated tasks in geotechnical engineering is examined here. In this study, first we clarify the necessary steps in using KH algorithm to automate the design procedure. Then, we explain all the effective parameters and limitations for handling the retaining wall problems which come from standard codes. As can be seen in Section 18.4, KH deals with this problem successfully and reached solutions.

References

1. A.H. Gandomi, A.H. Alavi. "Krill herd: a new bio-inspired optimization algorithm". *Communications in Nonlinear Science and Numerical Simulation*, vol. 17(12), pp. 4831-4845, 2012.
2. G.G Wang, L. Guo, A.H. Gandomi, G.S. Hao, H. Wang. "Chaotic krill herd algorithm". *Information Sciences*, vol. 274, pp. 17-34, 2014.
3. G. Wang, L. Guo, A.H. Gandomi, L. Cao, A.H. Alavi, H. Duan, J. Li. "Lévy-flight krill herd algorithm". *Mathematical Problems in Engineering*, Volume 2013, Article ID 682073, 14 pages, Hindawi 2013.
4. G.G. Wang, A.H. Gandomi, A.H. Alavi, S. Deb. "A multi-stage krill herd algorithm for global numerical optimization". *International Journal on Artificial Intelligence Tools*, vol. 25(2), 1550030, 2016.
5. G.G Wang, A.H. Gandomi, A.H. Alavi. "Stud krill herd algorithm". *Neurocomputing*, vol. 128, pp. 363-370, 2014.

6. J. Li, Y. Tang, C. Hua, X. Guan. "An improved krill herd algorithm: Krill herd with linear decreasing step". *Applied Mathematics and Computation*, vol. 234, pp. 356-367, 2014.
7. G.G. Wang, A.H. Gandomi, A.H. Alavi. "An effective krill herd algorithm with migration operator in biogeography-based optimization". *Applied Mathematical Modelling*, vol. 38(9-10), pp. 2454-2462, 2014.
8. A. Saribas, F. Erbatur. "Optimization and sensitivity of retaining structures". *Journal of Geotechnical Engineering*, vol. 122(8), pp. 649-656, 1996.
9. A.H. Gandomi, A.R. Kashani, D.A. Roke, M. Mousavi. "Optimization of retaining wall design using recent swarm intelligence techniques". *Engineering Structures*, vol. 103, pp. 72-84, 2015.
10. A.H. Gandomi, A.R. Kashani, D.A. Roke, M. Mousavi. "Optimization of retaining wall design using evolutionary algorithms". *Structural and Multidisciplinary Optimization*, vol. 55(3), pp. 809-825, 2017.
11. American Concrete Institute. Building code requirements for structural concrete and commentary (ACI 318-05), Detroit, 2005.