

# Solution for a Kind of Dynamic Optimization Based on Improved Krill Herd Algorithm

Xingjun Yuan<sup>1,\*</sup>, Yulei Ge<sup>2</sup> and Hao Lu<sup>1</sup>

<sup>1</sup>Systems Engineering Research Institute, Beijing 100094, China

<sup>2</sup>College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China

\*Corresponding author

**Abstract**—Considering a kind of dynamic optimization, an improved krill herd (KH) algorithm which is called GSA-KH is proposed in this paper. The improvements consist of three parts: 1) a good point set is constructed to obtain the initial krill population which can promote the representativeness of the initial population; 2) the speed factor is updated according to the changes of the krill population to accelerate the convergence; 3) an adaptive Cauchy mutation is employed so that the algorithm can escape the local optimum reasonably. Simulations on six test functions illustrate that the convergence and accuracy of GSA-KH algorithm are increased greatly than standard KH algorithm. Then the proposed algorithm is applied to solving a dynamic optimization cases.

**Keywords**—dynamic optimization; krill herd algorithm; adaptive cauchy mutation; CVP method; good point sets method; speed factor

## I. INTRODUCTION

Considering a typical dynamic optimization problem of continuous process as follows [1]:

$$\begin{aligned} \min J(u) &= \phi(x(t_f), t_f) + \int_0^{t_f} \psi(x(t), u(t), t) dt, \\ \text{s.t. } \frac{dx}{dt} &= f(x(t), u(t), t), \\ x(0) &= x_0, \\ u_{\min} &\leq u(t) \leq u_{\max}. \end{aligned} \quad (1)$$

Generally, there are two types of methods for solving the problem: direct methods and indirect methods. Direct methods convert an infinite dimensional dynamic optimization problem into a finite dimensional nonlinear programming problem through discretization [2-4]. Indirect methods use maximum principle to derive the necessary conditions for optimal control by constructing Hamiltonian function, and convert the original problem into a two-point boundary value problem. The gradient computation is the key to solve this optimization. But because of the dependence on initial points and huge cost for solving gradient, these methods usually consume large running time or even cannot obtain the optimal solution.

In recent years, the intelligent optimization algorithms develop rapidly because of its simple principle, good

robustness and powerful global searching ability. These algorithms are widely used in various types of optimization problems. Pham [6] proposed an evolutionary algorithm for solving the dynamic optimization of chemical processes, and the algorithm enhances population diversity by introducing a new breeding operation. Rajish, et al. [7] successfully used ant colony optimization (ACO) algorithm with region-reduction strategy to solve dynamic optimization problems. In 2012, Gandomi and Alavi proposed krill herd (KH) algorithm [8] for global optimization problem. They compared KH algorithm with eight well-known methods, and found that the KH algorithm was capable of efficiently solving a wide range of benchmark optimization problems. As a new algorithm, there are some papers about the improvement of KH algorithm [9-11], but applications of the algorithm are seldom.

In this paper, an improved KH algorithm named GSA-KH algorithm is proposed for solving a kind of dynamic optimization problem as equation (1). Compared with standard KH algorithm, the new one makes three points of improvements. Good point set is used to set initial population, speed factor is adjusted according to the changes of krill population, and adaptive Cauchy mutation is adopted in order to avoid falling into local optimum. Then GSA-KH algorithm is tested by several test functions. Finally, the proposed algorithm is applied to solving a chemical dynamic optimization problem.

## II. PROBLEM TRANSFORMATION

The control vector parameterization (CVP) method utilizes finite number of parameters to approximate control variable  $u(t)$  ( $t \in [t_0, t_f]$ ) which is changing with time continuously [12]. Time interval is divided into  $n$  subintervals  $[t_{k-1}, t_k]$  ( $k=1, 2, \dots, n$ ). The control variable in every subinterval is represented by a simple polynomial which can be determined by finite number of parameters, so the control variable can be denoted as follows:  $u(t) = \sum_{k=1}^n \phi_k(\theta_k)$ , where  $\phi_k$  is a simple polynomial,  $\theta_k$  is a finite dimensional parameter vector. And equation (1) can be rewritten as follows:

$$\begin{aligned} \min J(u) &= \phi(x(t_f), t_f) + \int_0^{t_f} \psi(x(t), \sum_{k=1}^n \phi_k(\theta_k), t) dt, \\ \text{s.t. } \frac{dx}{dt} &= f(x(t), \sum_{k=1}^n \phi_k(\theta_k), t), \\ x(0) &= x_0, \\ u_{\min} &\leq \phi_k(\theta_k) \leq u_{\max}, \end{aligned} \quad (2)$$

where  $J$  and  $f$  are both implicit functions of  $\theta_k$ .

### III. IMPROVED KRILL HERD ALGORITHM

#### A. Standard KH Algorithm

KH algorithm is proposed according to the herding behavior of krill individuals. The positions of krill individuals are determined by three main actions [13]:

i. Movement induced by other krill individuals; ii. Foraging activity; iii. Random diffusion.

The moving process of krill individuals can be described with Lagrangian model as below:

$$\frac{dX_i}{dt} = N_i + F_i + D_i, \quad (3)$$

where  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  represents the position vector of  $i$ th ( $i \in 1, 2, \dots, n$ ) krill;  $D$  is the dimension of search space,  $n$  is the number of krill population;  $N_i$  is the motion induced by other krill individuals,  $F_i$  is the foraging motion, and  $D_i$  is the random physical diffusion.

##### 1) Movement induced by other krill individuals

The direction of motion induced is estimated from a target swarm density (target effect), the local swarm density (local effect), and a repulsive swarm density (repulsive effect). So the movement induced by other krill individuals can be formulated as follows:

$$N_i^{\text{new}} = N^{\max} (\alpha_i^{\text{local}} + \alpha_i^{\text{target}}) + \omega_n N_i^{\text{old}}, \quad (4)$$

where  $N^{\max}$  is the maximum speed,  $\omega_n$  is the inertia motion weight in range  $[0, 1]$ ,  $\alpha_i^{\text{local}}$  represents the local effect,  $\alpha_i^{\text{target}}$  represents the target effect, and  $N_i^{\text{old}}$  represents the last motion induced.

##### 2) Foraging activity

The foraging motion is determined by two factors: the estimated food location and the previous experience about the

foraging location of itself. This motion can be described as follows:

$$F_i^{\text{new}} = V_f (\beta_i^{\text{food}} + \beta_i^{\text{best}}) + \omega_f F_i^{\text{old}}, \quad (5)$$

where  $V_f$  is the maximum foraging speed,  $\omega_f$  is the inertia weight of the foraging motion in the range of  $[0, 1]$ ,  $\beta_i^{\text{food}}$  represents the effect of food,  $\beta_i^{\text{best}}$  represents the effect of previous best foraging location of itself, and  $F_i^{\text{old}}$  represents the last foraging motion.

##### 3) Random diffusion

The physical diffusion is considered as a random process. The motion can be expressed as follows:

$$D_i = D^{\max} \delta, \quad (6)$$

where  $D^{\max}$  is the maximum diffusion speed,  $\delta$  is the random directional vector and its arrays are random values in the range of  $[-1, 1]$ . The motion is described as follows:

$$D_i = D^{\max} \left(1 - \frac{I}{I_{\max}}\right) \delta, \quad (7)$$

where  $I_{\max}$  is the maximum generation.

##### 4) Motion process of the KH algorithm

In general, the defined motions frequently change the position of a krill individual toward the global optimum. The foraging motion and the motion induced by other krill individuals contain two global and two local strategies. They are working in parallel which make KH a powerful algorithm. The position vector of a krill individual during interval  $t$  to  $t + \Delta t$  is formulated as follows:

$$X_i(t + \Delta t) = X_i(t) + V_i, \quad (8)$$

$$V_i = \Delta t \frac{dX_i}{dt}, \quad (9)$$

$$\Delta t = \frac{C_t}{D} \sum_{j=1}^D (UB_j - LB_j), \quad (10)$$

where  $\Delta t$  is speed factor, it decides the scale of search for the space,  $C_t$  is usually a constant between  $[0,2]$ ;  $UB_j$  and  $LB_j$  are upper and lower bounds of the  $j$ th variable.

In addition to the four parts above, genetic reproduction mechanisms are incorporated into the algorithm to improve the performance. And more detailed information can be referred to [8].

### B. GSA-KH Algorithm

In this part, the three points of improvements of KH algorithm are described in detail.

#### 1) Good point sets method

The definition of good point set [14] is expressed as follows:  $G_s$  is assumed as a  $D$ -dimensional unit cube in the Euclidean space, let  $P_n(k) = (\{r_1 * k\}, \dots, \{r_D * k\}), k = 1, 2, \dots, n, r \in G_s$ , (where  $\{r_i * k\}$  means the decimal part of it), if the deviation  $\varphi(n)$  of  $P_n(k)$  satisfies  $\varphi(n) = C(r, \varepsilon)n^{-1+\varepsilon}$  (where  $C(r, \varepsilon)$  is a constant related with  $r$  and  $\varepsilon$ ), then  $P_n(k)$  is called a good point set, and  $r$  is called a set of good points. The process is: let  $r_k = \{2\cos(2\pi k / p), 1 \leq k \leq D\}$ , if  $p$  is the minimum prime number of the numbers satisfying  $p - 3 \geq 2D$ , then  $r$  is a set of good points, and a good point set can be found.

A two-dimensional initial population with 100 individuals is generated by good point sets method. As shown in figure 1 and figure 2, it is obvious that the initial population generated by good point sets method is more even. So good point sets method is a nice way to generate initial populations.

The map from a  $D$ -dimensional unit cube to the search space is defined as follows:

$$f(r_j * k) = LB_j + \{r_j * k\} * (UB_j - LB_j), \quad (11)$$

where  $k = 1, 2, \dots, n$ ,  $n$  is the number of initial population,  $j = 1, 2, \dots, D$ ,  $D$  is the dimension of the search space,  $f(r_j * k)$  represents the  $j$ th variable of  $k$ th krill individual.

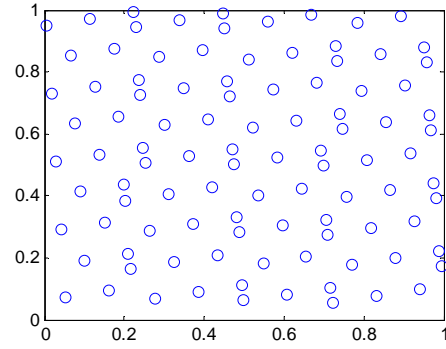


FIGURE I. THE INITIAL POPULATION GENERATED BY A GOOD POINT SET

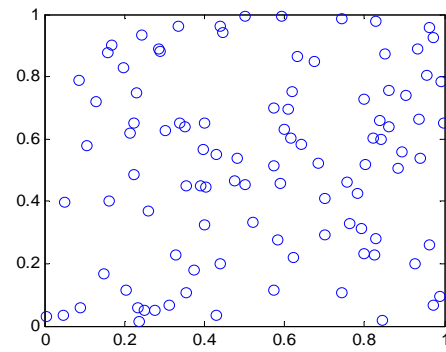


FIGURE II. THE INITIAL POPULATION GENERATED BY A RANDOM WAY

#### 2) Speed factor

In KH algorithm, the speed factor can be considered as a remarkable advantage of the algorithm in comparison with other intelligent algorithms. It determines how carefully the space is searched. The adjustment is denoted as follows:

$$lb_j = \min(\{X_{ij}, i = 1, 2, \dots, n\}), \quad (12)$$

$$ub_j = \max(\{X_{ij}, i = 1, 2, \dots, n\}), \quad (13)$$

$$\Delta t = \frac{C_t}{D} \sum_{j=1}^D (ub_j - lb_j). \quad (14)$$

In this way, the speed factor adjusts with the changes of  $lb$  and  $ub$ , who update according to the changes of krill population. It is worth noting that the adjustment doesn't change the lower and upper bounds of the variables, which means  $lb$  and  $ub$  don't replace  $LB$  and  $UB$ .

### 3) Adaptive Cauchy mutation

KH algorithm is lack of effective mutation mechanism for avoiding local optimum. The traditional mutation operator is Gaussian mutation which is formulated as follows:

$$X_{ij} = X_{ij} + \eta * G(0,1), \quad (15)$$

where  $i=1,2,\dots,n$ ,  $j=1,2,\dots,D$ ,  $\eta$  is a constant which controls the step of mutation,  $G(0,1)$  is a random number generated by Gaussian distribution with zero mean and unit variance.

A fast evolutionary programming algorithm is proposed in [15]. The proposed algorithm uses Cauchy mutation operator instead of Gaussian mutation operator.

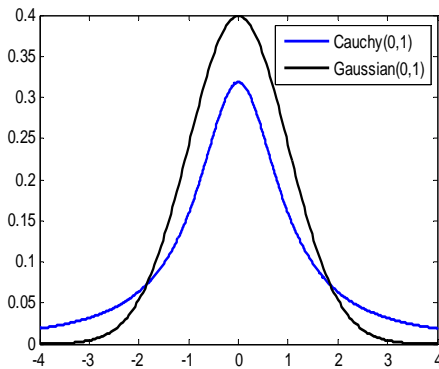


FIGURE III. COMPARISON BETWEEN GAUSSIAN AND CAUCHY FUNCTIONS

It is obvious from figure 3 that Cauchy mutation has a higher probability of escaping from a local optimum or moving away from a plateau. Cauchy mutation spends enough time in exploiting the local neighborhood of current location, avoids the blindness of random mutation. So Cauchy mutation is introduced to the improved algorithm.

The average speed of population is usually defined as follows:

$$\bar{V}_j = (\sum_{i=1}^n V_{ij}) / n. \quad (16)$$

where  $V_{ij}$  is the  $j$ th dimensional speed component of  $i$ th krill,  $\bar{V}_j$  is the average speed of  $j$ th dimension.

The formula of adaptive Cauchy mutation is denoted as follows:

$$X_{ij} = X_{ij} + \bar{V}_j * C(0,1), \quad (17)$$

where  $C(0,1)$  is a random number generated by Cauchy distribution with zero mean and unit variance.

### A. Main Procedure of the Algorithm

After the addition of three points of improvements, the main procedure of the algorithm is described as follows:

Step 1: Determine the values of  $n, D, I_{max}, \omega_n, \omega_f, N_{max}, V_f, D^{max}$  and the maximum number of mutation  $M$ .

Step 2: Generate the initial krill population with good point sets method.

Step 3: Evaluate each krill individual, find the krill with best fitness value, and let  $I = 1$ .

Step 4: Execute three motion operators and crossover operator.

Step 5: Update the krill individual, evaluate each krill individual and find the krill with best fitness value again.

Step 6: Calculate the average speed of the krill population, and let  $k = 1$ .

Step 7: Execute the adaptive Cauchy mutation at the current best krill (best fitness value). Compare the new krill after mutation with the current best krill, if it is better, replace the current best krill.

Step 8: Let  $k = k + 1$ , if  $k \leq M$ , return to Step 7, else go to Step 9.

Step 9: Let  $I = I + 1$ , if  $I \leq I_{max}$ , return to Step 3, else end the algorithm.

### B. Algorithm Test

In order to verify the performance of GSA-KH algorithm, six test functions are introduced to test the improved algorithm [16]. The specific test functions are listed in table 1. The standard KH algorithm is used to compare and analyze the performance.

TABLE I. MEAN BEST FITNESS VALUES AND STANDARD DEVIATIONS

Function	Function name	Search space	Optimal value	KH		GSA-KH	
				mean	std	mean	std
$f_1$	Rosenbrock	[-2.048,2.048]	0	8.68e+1	5.96e+1	2.52e+1	5.02e+1
$f_2$	Schwefel1.2	[-100,100]	0	5.73e+3	3.22e+3	1.41e+3	3.29e+3
$f_3$	Quartic	[-1.28,1.28]	0	9.34e-2	7.85e-2	4.13e-2	2.88e-2
$f_4$	Schwefel2.4	[0,10]	0	8.61e+1	1.77e+1	1.46e-5	9.26e-6
$f_5$	Ackley	[-32,32]	0	1.47e+0	8.04e-1	1.14e-3	6.85e-3
$f_6$	Griewank	[-600,600]	0	1.27e-1	2.38e-1	5.84e-2	1.73e-2

The parameters setting of GSA-KH algorithm is almost the same with that of KH algorithm. The maximum speed of three motions is:  $N^{\max} = 0.01$ ,  $V_f = 0.02$ ,  $D^{\max} = 0.005$ .  $C_i$  is usually set as 0.5. The inertia weight ( $\omega_n, \omega_f$ ) is equal to 0.9 at the beginning and decrease to 0.1 linearly with the iteration.

The maximum number of mutation  $M$  is determined according to the actual situation. A better result may be obtained with a bigger  $M$ , but more computational cost will be consumed at the same time. The number of krill population is 20, the dimension of krill individuals is 20, and the maximum generation is  $I_{\max} = 300$ .

In order to eliminate the randomness of results, the algorithm is carried out 30 trials. In table 1, the mean best fitness values and standard deviations are given. It can be seen that the convergence accuracy of GSA-KH algorithm is better than that of KH algorithm.

#### IV. CASE STUDIES

In this section, the GSA-KH algorithm is applied to solve a dynamic optimization case. The maximum number of mutation  $M$  is 20. The algorithm is carried out 20 times. And the state variables are computed by Runge-Kutta method.

Considering a first-order irreversible chemical reaction carried out in a continuous stirred tank reactor (CSTR) [5]. The optimal problem can be modelled as follows:

$$\min J(u) = \int_0^{t_f} (x_1^2 + x_2^2 + 0.1u^2) dt, \quad (18)$$

$$s.t. \frac{dx_1}{dt} = -(2+u)(x_1 + 0.25) + (x_2 + 0.5) \exp\left(\frac{25x_1}{x_1 + 2}\right), \quad (19)$$

$$\frac{dx_2}{dt} = 0.5 - x_2 - (x_2 + 0.5) \exp\left(\frac{25x_1}{x_1 + 2}\right), \quad (20)$$

$$t_f = 0.78, x(0) = [0.09 \ 0.09]^T, 0 \leq u \leq 5. \quad (21)$$

where  $x_1$  represents the deviation from dimensionless steady-state temperature and  $x_2$  is the deviation from the dimensionless steady-state concentration, the control variable  $u$  represents the manipulation of the flow-rate of the cooling fluid.

The time interval  $[0, t_f]$  is divided into 13 subintervals, the control variable in every subinterval is replaced with a constant. Then the optimal problem can be solved by the proposed algorithm, the best fitness value obtained is 0.135345. The optimum value obtained is 0.13309 [5].

The optimal control trajectory is shown in figure 4. And the curve of best fitness value versus number of generations is shown in figure 5, the curve gradually becomes flatly at about 70 generations.

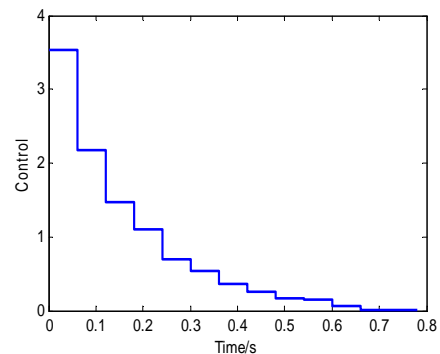


FIGURE IV. OPTIMAL CONTROL PROFILE

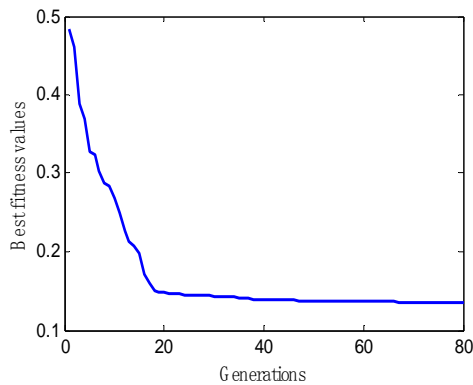


FIGURE V. BEST FITNESS VALUES VS. GENERATIONS

It can be seen from figure 5 that the algorithm arrives at a stable stage within 70 generations. This point shows that the algorithm has a nice convergence rate. To make a comparison with previous researches, a summary of the results is listed in table 2, where “Best” and optimum mean the best values ever obtained in previous researches and in this paper, respectively. FE represents function evaluations. According to table 2, it is obvious that GSA-KH algorithm performs well on both convergence accuracy and rate.

TABLE II. SUMMARY OF RESULTS

Case	Best	Method	Optimum	FE
Case 1	0.13309	EA [17]	0.135586	2270
		IDP [5]	0.13309	—
		<b>GSA-KH</b>	<b>0.135345</b>	<b>2800</b>

## V. CONCLUSION

In this paper, a novel method for dynamic optimization based on improved KH algorithm is proposed. Firstly, three points of improvements are added into the KH algorithm. An initial population is generated by good point sets method. Speed factor is adjusted with the changes of krill populations. Adaptive Cauchy mutation is used to increase the ability of the algorithm avoiding local optimum. Then the performance of GSA-KH algorithm is verified by several test functions. At last, combining with CVP method, the algorithm demonstrates its feasibility and robustness through the successful application to two dynamic optimization cases of chemical processes. It is no doubt that the proposed algorithm can be regarded as a reliable and efficient tool for solving this type of problems.

## REFERENCES

[1] S. Ghasemi, A. Nazemi, S. Hosseinpour, Nonlinear fractional optimal control problems with neural network and dynamic optimization schemes, *Nonlinear Dynamics*, 89(4): 2669-2682, 2017.

[2] L. T. Biegler, Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation, *Computers & Chemical Engineering*, 8(3): 243-247, 1984.

[3] M. A. Sychev, Variational field theory from the point of view of direct methods, *Siberian Mathematical Journal*, 58(5): 891-898, 2017.

[4] X. G. Liu, L. Chen, Y. Q. Hu, Solution of chemical dynamic optimization using the simultaneous strategies, *Chinese Journal of Chemical Engineering*, 21(1): 55-63, 2013.

[5] R. Luus, Optimal control by dynamic programming using systematic reduction in grid size, *International Journal of Control*, 51(5): 995-1013, 1990.

[6] Q. T. Pham, Dynamic optimization of chemical engineering processes by an evolutionary method, *Computers & Chemical Engineering*, 22(7/8): 1089-1097, 1998.

[7] Rajesh, K. Gupta, H. S. Kusumakar, et al., Dynamic optimization of chemical processes using ant colony framework, *Computers & Chemistry*, 25(6): 583, 2001.

[8] H. Gandomi, A. H. Alavi, Krill herd: A new bio-inspired optimization algorithm, *Communications in Nonlinear Science & Numerical Simulation*, 17(12): 4831-4845, 2012.

[9] G. G. Wang, A. H. Gandomi, A. H. Alavi, An effective krill herd algorithm with migration operator in biogeography-based optimization, *Applied Mathematical Modelling*, 38(9-10): 2454-2462, 2014.

[10] G. G. Wang, A. H. Gandomi, A. H. Alavi, Stud krill herd algorithm, *Neurocomputing*, 128: 363-370, 2014.

[11] G. G. Wang, A. H. Gandomi, A. H. Alavi, et al., Hybrid krill herd algorithm with differential evolution for global numerical optimization, *Neural Computing and Applications*, 25(2): 297-308, 2014.

[12] Schlegel, K. Stockmann, T. Binder, et al., Dynamic optimization using adaptive control vector parameterization, *Computers & Chemical Engineering*, 29(8): 1731-1751, 2005.

[13] E. E. Hofmann, A. G. E. Haskell, J. M. Klinck, et al., Lagrangian modelling studies of Antarctic krill (*Euphausia superba*) swarm formation, *ICES Journal of Marine Science: Journal du Conseil*, 61(4): 617-631, 2004.

[14] X. Xia, Z. X. Cai, Y. Wang, et al., Constrained optimization evolutionary algorithm based on good lattice points principle, *Control and Decision*, 24: 249-253, 2009.

[15] Narang, R. S. Patwal, H. Garg, A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units, *Energy*, 2017.

[16] M. Jamil, X. S. Yang, A literature survey of benchmark functions for global optimisation problems, *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2): 150-194, 2013.

[17] L. Lopez Cruz, L. G. Van Willigenburg, G. Van Straten, Efficient Differential Evolution algorithms for multimodal optimal control problems, *Applied Soft Computing*, 3(2): 97-122, 2003.