ISSN: 2688-9323

Logistics Distribution Center Location Problem with Cuckoo Search Algorithm

Hui Zhang, Jiayou Gong

School of Economics and Management, Wuchang Institute of Technology, Wuhan 430065, China

Abstract

Cuckoo search (CS) algorithm is a novel swarm intelligence optimization algorithm, which is successfully applied to solve some optimization problems. In this work, CS is introduced to solve Logistics distribution center location problem. Comparing with various IGA algorithms, the results showed that the CS approach obtained higher computational efficiency than IGA comparative methods in both 6 distribution centers and 10 distribution centers, which demonstrate CS is a competitive swarm algorithm.

Keywords

Global optimization; Cuckoo search algorithm; Logistics distribution center location.

1. Introduction

Optimization problems have been one of the most important research topics in recent years. They exist in many domains, such as scheduling, image processing, feature selection and detection, path planning, and test-sheet composition. Metaheuristic algorithms, a theoretical tool, are based on nature-inspired ideas, which have been extensively used to solve highly non-linear complex multi-objective optimization problems. A number of popular metaheuristics having a stochastic nature are compared in some literatures with deterministic Lipschitz methods by using operational zones. Most of these metaheuristics methods are inspired by natural or physical processes, such as bat algorithm (BA) [1], biogeography-based optimization (BBO) [2], ant colony optimization (ACO) [3], artificial bee colony (ABC) [4-6], harmony search (HS) [7, 8], monarch butterfly optimization (MBO) [9, 10], particle swarm optimization (PSO) [11, 12], genetic programming [13], immune genetic algorithm(IGA) [14], and cuckoo search (CS) [15-21].

Yang and Deb [19] proposed a metaheuristic optimization method, named CS algorithm which is inspired by smart incubation behavior of a type of birds called cuckoos in nature. CS performs local search well in most cases, but sometimes it may have no ability of escaping from local optima which restricts its ability to carry out full search globally. In order to enhance the ability of the CS, Mlakar et al. [22] proposed a novel hybrid self-adaptively CS algorithm adding three features, i.e., a self-adaptively of cuckoo search control parameters, a linear population reduction, and a balancing of the exploration search strategies. In addition, Li et al. [23] enhanced the exploitation ability of the cuckoo search algorithm by using an orthogonal learning strategy. An improved discrete version of CS was presented by Ouaarab et al. [24].

In this paper, we used CS algorithm to solve logistics distribution center location problem. The experimental results with the other approaches demonstrated the superiority of the proposed strategy. A series of simulation experiments show that CS performs more accurately and efficiently than other evolutionary methods in terms of the quality of the solution and convergence rate.

2. Cuckoo Search

The cuckoo search algorithm [19] is a stochastic optimization algorithm that models brood parasitism of cuckoo birds. The algorithm is based on the obligate brood parasitic behavior found in some cuckoo nests by combining a model of this behavior with the principles of Lévy flights, which discard worst solutions and generate new ones after some certain iteration.

According to the mentioned characteristics, CS can be expressed as three idealized rules:

1) Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.

2) The best nests with the highest-quality eggs (solutions) will be carried over to the next generations.

3) The number of available host nests is fixed, and the alien egg is discovered by the host bird with the probability, $P_a \in [0,1]$. If the alien egg is discovered, the nest is abandoned and a new nest is built in a new location.

The CS algorithm is equiponderant to the integration of a Lévy flights. The position of the number *i* nest are indicated by using *D*-dimensional vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{id}),\ 1 \le i \le n$, a Lévy flight is performed:

$$X_i^{t+1} = x_i^t + a \otimes levy(\lambda) \qquad (i = 1, 2, \ldots, n), \tag{1}$$

$$a = a_0 \otimes (x_j^t - x_i^t) \tag{2}$$

where $\alpha > 0$ is the step size that is used to control the range of the random search, which should be related to the scales of the problem of interests, and step size information is more useful can be computed by Eq. (2). The product $\otimes$ means entry-wise multiplications. x_i^t and x_j^t are two different solutions selected randomly. A new solution with the same number of cuckoos is generated after partial solutions are discarded. $levy(\lambda)$ with the random walk can be expressed in terms of a simple power-law equation.

$$levy(\beta) \sim \mu = t^{-1-\beta},\ 0 < \beta \le 2 \tag{3}$$

where μ and t are two random numbers following the normal distribution, β often takes a fixed value of 1.5.

$$levy(\beta) \sim \frac{\phi \times \mu}{|v|^{1/\beta}} \tag{4}$$

$$\phi = \left[\frac{\Gamma(1+\beta) \times \sin(\frac{\pi \times \beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \right]^{1/\beta} \tag{5}$$

where Γ is gamma function, μ and v are random numbers drawn from a normal distribution with mean of 0 and standard deviation of 1, which has an infinite variance with an infinite mean. Here the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step length distribution with a heavy tail. In Lévy flights random walk component, the new solution X_i is generated through Eq. (6).

$$X_{g+1,i} = X_{g,i} + \alpha_0 \frac{\phi \times \mu}{|v|^{1/\beta}} (X_{g,i-} X_{g,best}) \tag{6}$$

where $X_{g,best}$ represents the best solution obtained so far, α_0 is a scaling factor. The Lévy distribution is a process of random walk, after a series of smaller steps, Lévy flights can

ISSN: 2688-9323

suddenly obtain a relatively larger step size. Lévy distribution is implemented at the initial stage of algorithm, which helps to jump out of the local optimum.

Algorithm : *CS algorithm*

(1) randomly initialize population of n host nests
(2) calculate fitness value for each solution in each nest
(3) ***while*** *(stopping criterion is not meet do)*
(4) Generate x_i^{t+1} *as new solution by using Lévy flights;*
(5) Choose candidate solution x_i^t*;*
(6) ***if*** $f(x_i^t) > f(x_i^{t+1})$
(7) Replace x_i^t *with new solution* x_i^{t+1}*;*
(8) ***end if***
(9) Throw out a fraction (p_a) *of worst nests;*
(10) Generate solution k_i^{t+1} *using Eq. (3);*
(11) ***if*** $f(x_i^t) > f(x_i^{t+1})$
(12) Replace x_i^t *with new solution* x_i^{t+1}*;*
(13) ***end if***
(14) Rank the solution and find the current best.
(15) ***end while***

3. Application in the Problem of Logistics Distribution Center Location

3.1. Problem Description

The location of distribution center determines the efficiency of the entire logistics network system and the utilization of resources. The problem of logistics distribution center location can be described as: *m* cargo distribution center are search in *n* demand point, so that the distance between *m* searched distribution centers and other *n* cargo demand points is the shortest. At the same time, the following constraint conditions must be met: the supply of goods in the distribution center can meet the requirements of the cargo demand point; the goods required for a cargo demand point can only be provided by one distribution center; the cost of transporting the goods to the distribution center is not considered. According to the above assumptions, the mathematical model of the problem for logistics distribution center location can be described as:

$$\min(\cos t) = \sum_{i=1}^{m}\sum_{j=1}^{n}(need_j \cdot dist_{i,j} \cdot \mu_{i.j}) \tag{7}$$

$$s.t. \qquad \sum_{i=1}^{m}\mu_{i,j} = 1, i \in M, j \in N \tag{8}$$

$$\mu_{i,j} \le h_j, i \in M, j \in N \tag{9}$$

$$\sum_{t=1}^{m} h_i \le p, i \in M \tag{10}$$

$$h_j \in \{0,1\}, i \in M \tag{11}$$

$$\mu_{i,j} \in \{0,1\}, i \in M, j \in N \tag{12}$$

$$M = \{ j \mid j = 1, 2, \ldots, m \} \qquad N = \{ j \mid j = 1, 2, \ldots, n \} \tag{13}$$

where Eq. (7) is the objective function, *cost* represents the transportation cost, m is the number of logistics distribution center, $nest_j$ is the demand quantity of demand point j, $dist_{i,j}$ indicate the distance between distribution center i and goods demand point j. When $u_{i,j}$ is equal to 1, the goods of demand point j are distributed by distribution point i. Eqs. (8)-(13) are the constraints.

3.2. Analysis of Experimental Results

In order to verify the performance of the DMQL-CS algorithm in solving the problem of logistics distribution center location, 40 demand points were adopted. The geographical position coordinates and demands were shown in Table 1. All the experiments were carried out on a P4 Dual-core platform with a 1.75 GHz processor and 4 GB memory, running under the Windows 7.0 operating system. The algorithms were written by MATLAB R2017a. The maximum number of iterations, population size, and the times of running were set to 30,000, 15, and 30, respectively.

Table 1. The geographical position coordinates and demands

No	coordinates		demand	No	coordinates		demand	No	coordinates		demand	No	coordinates		demand
	x	y			x	y			x	y			x	y	
1	97	28	94	11	91	96	85	21	111	117	92	31	125	66	45
2	100	56	11	12	39	90	54	22	63	42	99	32	169	49	98
3	45	67	50	13	50	101	25	23	67	105	98	33	31	188	31
4	150	197	88	14	67	66	87	24	160	156	88	34	86	42	91
5	105	48	80	15	157	54	66	25	100	125	47	35	90	21	79
6	24	158	29	16	104	35	82	26	35	48	47	36	46	53	47
7	88	61	93	17	169	95	48	27	143	172	34	37	62	30	84
8	55	105	10	18	48	39	78	28	94	56	33	38	163	176	52
9	120	120	18	19	115	61	16	29	57	73	43	39	190	141	10
10	43	105	38	20	154	174	49	30	25	127	100	40	170	30	77

In order to further verify the efficiency of the CS algorithm, in this section, the effectiveness of the proposed method is verified by comparing the cuckoo search algorithm (CS) and the immune genetic algorithm (IGA). Fig. 1 shows the average convergence curve and optimal convergence curve of CS algorithm for running 20 times, 30 times and 50 times respectively in 40 cities and 6 distribution centers. The 6 optimal distribution center points and optimal routes are found by CS algorithm also is shown in Fig. 1. Fig. 2 shows the average convergence curve and optimal convergence curve of CS algorithm for running 20 times, 30 times and 50 times respectively in 40 cities and 10 distribution centers. Table 2 and Table 3 show distribution ranges two algorithms (CS and IGA) for 6 and 10 distribution centers in 40 cities, respectively.

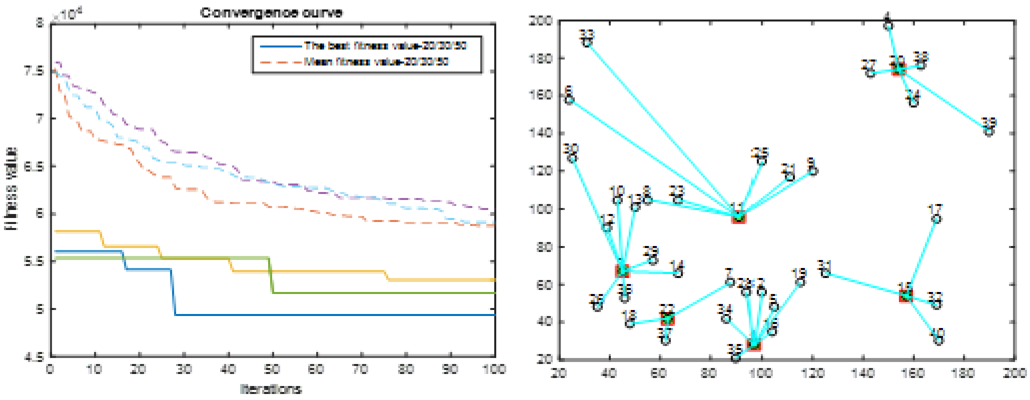

Fig. 1. Convergence curves and optimal distribution centers scheme for the CS algorithm in 6 distribution centers

For the first set of experiments, the CS algorithm is run 20, 30 and 50 times independently in 40 cities 6 distribution center. It can be seen from Fig. 1, the average convergence curve can converge at 30 iterations. It indicates that the fitness value decreases rapidly for the logistics distribution center location method based on CS algorithm at early stage of the algorithm. The optimal distribution cost and average distribution cost obtained by the CS algorithm are 4.9629E+04, which indicates that CS has high solution accuracy in 6 distribution centers and reduces the cost of logistics distribution. The optimal distribution center points found in Fig. 1 are: 3, 11, 22, 1, 15, and 20.

For the second set of experiments, the CS algorithm is run 20, 30 and 50 times independently in 40 cities 10 distribution center as shown in Fig. 2. The optimal distribution cost and average distribution cost obtained by the CS algorithm are 3.2435E+04, which indicates that CS has high solution accuracy in 10 distribution centers and reduces the cost of logistics distribution. The optimal distribution center points found in Fig. 2 are: 6, 8, 18, 11, 21, 28, 16, 1, 20, and 15.

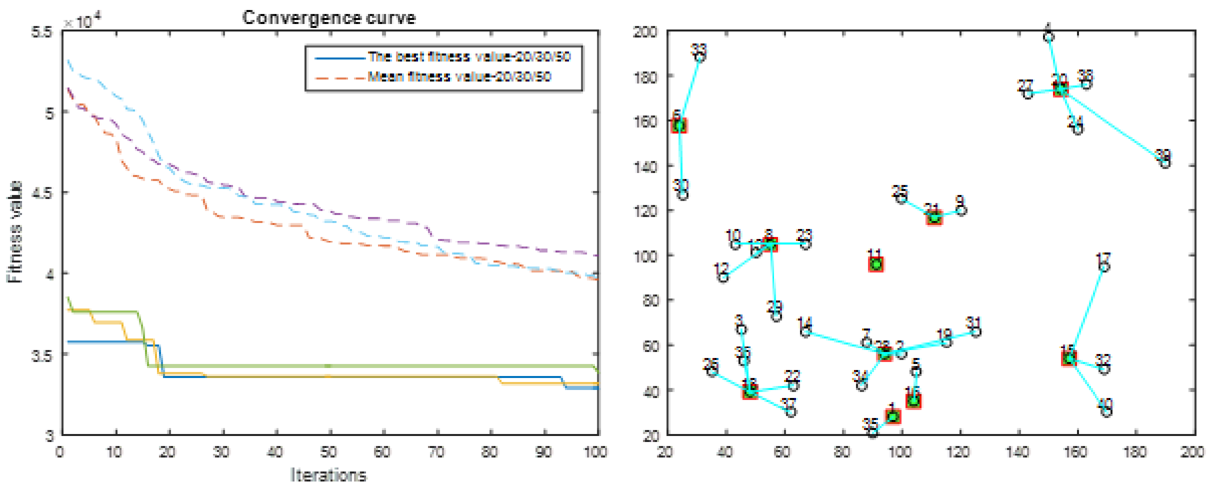

Fig. 2. Convergence curves and optimal distribution centers scheme for the CS algorithm in 10 distribution centers

Table 2. The distribution scheme two algorithms (CS and IGA) for 6 distribution centers in 40 city

CS		IGA	
D-C	Distribution scope	D-C	Distribution scope
3	30, 12, 10, 13, 29, 14, 36, 26	10	33, 6, 30, 12, 8, 23, 13
11	8, 23, 6,33, 25, 21, 9	22	26, 36, 3, 18, 29, 14, 37, 35
22	18, 37, 7	21	25, 11, 9
1	34, 35, 28, 2, 5, 16, 19	2	7, 34, 28, 19, 31, 1, 16, 5, 19
15	31, 32, 17, 40	20	4, 27, 24, 38, 39
20	27, 4, 38, 24, 39	17	15, 32, 40

Table 3. The distribution scheme two algorithms (CS and IGA) for 10 distribution centers in 40 city

CS		IGA	
D-C	Distribution scope	D-C	Distribution scope
6	30, 33	30	6, 33
8	10, 12, 13, 23, 19	23	12, 10, 13, 8
18	3, 26, 36, 22, 37	14	29, 3, 26, 36, 18, 22
11	-	1	34, 37, 35, 16
21	25, 9	2	7, 28,5,19,31
28	14, 7, 34, 2, 19, 31	11	-
16	5	25	21, 9
1	35	24	17, 20, 38, 39
20	4, 27, 38, 24, 39	15	17, 32, 40
15	17, 32, 40	4	-

Due to limited space, only one comparison algorithms (IGA) is listed in this paper. IGA algorithm introduced crossover and variation strategy into immune algorithm, which improves performance of the immune algorithm. In this experiment, the convergence curves and optimal distribution scheme diagrams of 6 distribution centers and 10 distribution centers in 40 cities are shown respectively. The 6 optimal distribution centers and distribution addressing schemes for these algorithms are shown in Table 2, and the 10 optimal distribution centers and distribution addressing schemes are shown in Table 3.

For the third set of experiments, Although the IGA algorithm can converge, it has a lot of noise for the average convergence curve. The convergence effect of IGA is worse compared with CS algorithm. The optimal distribution centers and distribution addressing schemes are shown in Tables 2 and 3. According to Tables 2 and 3, the optimal distribution center points found by CS algorithm for 6 and 10 distribution centers are (3, 11, 22, 1, 15, 20) and (6, 8, 18, 11, 21, 28, 16, 1, 20, 15). The optimal distribution center points found by IGA algorithm for 6 and 10 distribution centers are (10, 22, 21, 2, 20, 17) and (30, 23, 14, 1, 2, 11, 25, 24, 15, 4).

4. Conclusion

In this paper, we constructed a CS model to solve the address of logistics distribution center. The results showed that the CS algorithm clearly outperformed the standard IGA algorithm. Comparing with IGA, CS has achieved good results in both 6 distribution centers and 10 distribution centers.

In the future, we will focus our research work on the study of special cases to strengthen the algorithm in more complex conditions. We will determine how to generalize our work to handle combinatorial optimization problems and to extend CS optimization algorithms to in the realistic engineering areas and feature selection for machine learning.

Acknowledgements

This work was supported by the scientific research project of Hubei Provincial Department of Education (B2018327).

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

[1] X. S. Yang, and A. H. Gandomi, “Bat algorithm: a novel approach for global engineering optimization,” Engineering Computations, vol. 29, no. 5, pp. 464-483, 2012.

[2] D. Simon, “Biogeography-based optimization,” IEEE Transactions on Evolutionary Computation, vol. 12, no. 6, pp. 702-713, 2008.

[3] M. Dorigo, and T. Stutzle, Ant Colony Optimization, MIT Press, Cambridge, 2004.

[4] H. Wang, and J.-H. Yi, “An improved optimization method based on krill herd and artificial bee colony with information exchange,” Memetic Computing, vol. 10, no. 2, pp. 177-198, 2018.

[5] D. Karaboga, and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” Journal of Global Optimization, vol. 39, no. 3, pp. 459-471, 2007.

[6] F. Liu, Y. Sun, G.-G. Wang, and T. Wu, “An artificial bee colony algorithm based on dynamic penalty and chaos search for constrained optimization problems,” Arabian Journal for Science and Engineering, vol. 43, no. 12, pp. 7189-7208, 2018.

[7] Z. W. Geem, J. H. Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: harmony search,” Simulation, vol. 76, no. 2, pp. 60-68, 2001.

[8] G.-G. Wang, A. H. Gandomi, X. Zhao, and H. E. Chu, “Hybridizing harmony search algorithm with cuckoo search for global numerical optimization,” Soft Computing, vol. 20, no. 1, pp. 273-285, 2016.

[9] G.-G. Wang, S. Deb, and Z. Cui, “Monarch butterfly optimization,” Neural Computing and Applications, vol. 31, no. 7, pp. 1995-2014, 2019.

[10] G.-G. Wang, S. Deb, X. Zhao, and Z. Cui, “A new monarch butterfly optimization with an improved crossover operator,” Operational Research: An International Journal, vol. 18, no. 3, pp. 731-755, 2018.

[11] J. Kennedy, and R. Eberhart, "Particle swarm optimization." pp. 1942-1948.

[12] Y. Sun, L. Jiao, X. Deng, and R. Wang, “Dynamic network structured immune particle swarm optimisation with small-world topology,” International Journal of Bio-Inspired Computation, vol. 9, no. 2, pp. 93-105, 2017.

[13] A. H. Gandomi, and A. H. Alavi, “Multi-stage genetic programming: A new strategy to nonlinear system modeling,” Information Sciences, vol. 181, no. 23, pp. 5227-5239, 2011.

[14] X. Z. Wang, X.-j.; Cao, X.; Zhang, J.; Fen, L, “An improved genetic algorithm based on immune principle,” Minimicro Systems, vol. 20, no. 117, pp. 120, 1999.

[15] J. Li, Y.-x. Li, S.-s. Tian, and J. Zou, “Dynamic cuckoo search algorithm based on Taguchi opposition-based search,” International Journal of Bio-Inspired Computation, vol. 13, no. 1, pp. 59-69, 2019.

[16] X. S. Yang, and S. Deb, “Engineering optimisation by cuckoo search,” International Journal of Mathematical Modelling and Numerical Optimisation, vol. 1, no. 4, pp. 330-343, 2010.

[17] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,” Engineering with Computers, vol. 29, no. 1, pp. 17-35, 2013.

[18] G.-G. Wang, S. Deb, A. H. Gandomi, Z. Zhang, and A. H. Alavi, “Chaotic cuckoo search,” Soft Computing, vol. 20, no. 9, pp. 3349-3362, 2016.

[19] X.-S. Yang, and S. Deb, "Cuckoo search via Lévy flights." pp. 210-214.

[20] G.-g. Wang, Y.-x. Li, C. Liu, T. Zhang, D.-d. Xiao, and J. Li, “Multi-Swarm Cuckoo Search Algorithm with Q-Learning Model,” The Computer Journal, 2020.

[21] J. Li, Y.-x. Li, S.-s. Tian, and J.-l. Xia, “An improved cuckoo search algorithm with self-adaptive knowledge learning,” Neural Computing and Applications, 2019.

[22] U. Mlakar, I. Fister, and I. Fister, “Hybrid self-adaptive cuckoo search for global optimization,” Swarm and Evolutionary Computation, vol. 29, pp. 47-72, 2016.

[23] X. Li, J. Wang, and M. Yin, “Enhancing the performance of cuckoo search algorithm using orthogonal learning method,” Neural Computing and Applications, vol. 24, no. 6, pp. 1233-1247, 2013.

[24] A. Ouaarab, B. Ahiod, and X.-S. Yang, “Discrete cuckoo search algorithm for the travelling salesman problem,” Neural Computing and Applications, vol. 24, no. 7-8, pp. 1659-1669, 2013.