Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2843436, 11 pages
https://doi.org/10.1155/2020/2843436

Research Article

GIS-Based Niche Hybrid Bat Algorithm for Solving Optimal Spatial Search

Guoming Du,[1] **Yangbo Chen**,[1] **and Wei Sun**[1,2]

[1]*School of Geography and Planning, Sun Yat-sen University, 135 West Xingang RD., Guangzhou 510275, China*
[2]*Southern Marine Science and Engineering, Guangdong Laboratory, Zhuhai 519082, China*

Correspondence should be addressed to Yangbo Chen; eescyb@mail.sysu.edu.cn

Received 29 December 2019; Revised 19 March 2020; Accepted 8 April 2020; Published 11 May 2020

Academic Editor: Ricardo Aguilar-Lopez

Complex nonlinear optimization problems are involved in optimal spatial search, such as location allocation problems that occur in multidimensional geographic space. Such search problems are generally difficult to solve by using traditional methods. The bat algorithm (BA) is an effective method for solving optimization problems. However, the solution of the standard BA is easily trapped at one of its local optimum values. The main cause of premature convergence is the loss of diversity in the population. The niche technique is an effective method to maintain the population diversity, to enhance the exploration of the new search domains, and to avoid premature convergence. In this paper, a geographic information system- (GIS-) based niche hybrid bat algorithm (NHBA) is proposed for solving the optimal spatial search. The NHBA is able to avoid the premature convergence and obtain the global optimal values. The GIS technique provides robust support for processing a substantial amount of geographical data. A case in Fangcun District, Guangzhou City, China, is used to test the NHBA. The comparative experiments illustrate that the BA, GA, FA, PSO, and NHBA algorithms outperform the brute-force algorithm in terms of computational efficiency, and the optimal solutions are more easily obtained with NHBA than with BA, GA, FA, and PSO. Moreover, the precision of NHBA is higher and the convergence of NHBA is faster than those of the other algorithms under the same conditions.

1. Introduction

The general goal of the optimal spatial search is to find a set of contiguous places that meet specific optimization objectives, such as minimizing the total cost and maximizing the proximity to certain facilities [1]. Optimal spatial search problems often exhibit considerable complexity, especially when they are involved in large, multiple objective location allocation problems, many of which contain combinatorial optimization under complicated, highly nonlinear constraints, and thus, they are often NP-hard [2, 3]. The traditional solutions include the fixed charge location model, the covering model in noncompetitive location theory, and the medianoid and centroid models from competitive location theory. Because of the exponential increase in the search space with problem size, the combinatorial optimal problems where the reliability goal is achieved by discrete choices made from available parts encounter computational difficulties. Therefore, the methods are difficult to run in the acceptable computational time. Consequently, many research efforts have been devoted to improving or finding efficient algorithms. Metaheuristic algorithms, derived from the behavior of biological or physical systems found in nature, have become powerful methods to solve the tough spatial optimization problems [4]. Inspired by this idea, the genetic algorithm (GA) [5], ant colony optimization (ACO) [6], particle swarm optimization (PSO) [7], artificial fish swarm algorithm (AFSA) [8], cuckoo search (CS) [9], monkey algorithm (MA) [10], and firefly algorithm (FA) [11] have been proposed and applied widely. Every algorithm has its own advantages and disadvantages and is also suited for solving different problems. The bat algorithm (BA) is also a metaheuristic algorithm inspired by a property known as echolocation, which is a type of sonar that guides bats in their flying and hunting behavior [12]. Not only can the bats move, but also they can distinguish different types of insects even in complete darkness, thanks to such capabilities [12, 13]. The BA can solve optimization problems completely stronger than the PSO, GA, and harmony search algorithms [14]. The main reason is that the BA uses an appropriate combination of significant advantages of the mentioned algorithms.

However, the solution of the BA is sometimes trapped at one of its local optima because of its premature convergence and weak exploitation capabilities. Therefore, it cannot always obtain the ideal results when BA is used alone [15]. The main cause of premature convergence is the loss of diversity in the population. The niche techniques are regarded as an effective method to maintain the population diversity to enhance the exploration of the new search domains [16]. The niche techniques have been proven particularly useful in problems that require multiple solutions to be found by a search algorithm, such as multimodal and multiobjective optimization problems [17]. As a consequence, the aim of this study is to propose the niche hybrid bat algorithm (NHBA) to enhance the performance of BA on nonlinear optimization problems, avoid becoming trapped in a locally optimal value with the subsequent premature convergence problem of BA, and solve the complex problems of optimal spatial search more efficiently, precisely, and reliably.

The main contributions of this study are as follows:

(1) It provides a robust solution by hybridizing the niche techniques with BA in combination with GIS techniques

(2) It contributes to maintaining the diversity of the population properly, improving the global search ability, and overcoming the deficiency in the premature convergence on only one solution

(3) It provides an effective method for solving optimal spatial search and multiobjective nonlinear optimization problems

The rest of the paper is structured as follows. Section 2 gives the literature review of the related works. Section 3 introduces the basic concepts, theory, and procedure of BA. Section 4 proposes the GIS-based niche hybrid bat algorithm (NHBA). Section 5 analyzes and discusses the experimental results by implementing comparative studies. Section 6 concludes the paper.

2. Literature Review

Optimal spatial search is involved in combinatorial optimization problems such as location allocation. The genetic algorithm (GA), which searches over regions, was first proposed to identify a final, feasible, optimal, or near-optimal solution to a relaxed version of the redundancy allocation problem [18]. Because GA involves extensive coding and decoding, it will take an excessive amount of time to compute. Simulated annealing was used to solve high-dimensional nonlinear optimization problems for multisite land use allocation (MLUA) problems [2]. Simulated annealing can almost guarantee finding the optimal solution when the cooling process is slow enough, and the simulation is running long enough. However, the fine adjustment of parameters affects the convergence rate of the optimization process. The multiobjective harmony search algorithm was used to optimize the multiserver location allocation problem in congested systems [19]. A modified fruit fly optimization algorithm (MFOA) was proposed to solve a location allocation-inventory problem in a two-echelon supply chain network [20]. Ant colony optimization (ACO) with stochastic demand was used to solve the stochastic uncapacitated location allocation problem with an unknown number of facilities [21]. A metaheuristic algorithm based on the harmony search algorithm was presented for the location, allocation, and routing of temporary health centers in rural areas in a crisis situation [22]. An improved artificial bee colony was proposed for the facility location allocation problem of the end-of-life vehicles recovery network [23]. A self-learning particle swarm optimization (SLPSO) algorithm was developed to solve the multiechelon capacitated location-allocation-inventory problem [24]. It is difficult to avoid becoming stuck at the local minimum value when using the single algorithm. Furthermore, these algorithms lack the support of GIS techniques.

The BA was proposed by Yang in 2010. There are many variants of BA in the related literature. A multiobjective bat-inspired algorithm was proposed for solving nonlinear global optimization problems [14]. Simulation results suggested that the proposed algorithm worked efficiently. Yang's original work was expanded to solve complicated constrained nonlinear optimization problems [25]. A modified optimization methodology called opposition-based BA (OBA) was proposed for the infinite impulse response system identification problem [26]. The simulation results revealed OBA to be a more competent candidate than other evolutionary algorithms (EA), such as real-coded GA, differential evolution (DE), and PSO, in terms of accuracy and convergence speed. The discrete real BA (RBA) was proposed for route search optimization of a graph-based road network [27]. Compared with ACO and the intelligent water drops (IWD) algorithm, RBA achieved a better convergence rate. A novel BA was developed in order to implement the loading pattern optimization of a nuclear reactor core [28]. The test results showed that BA was a very promising algorithm for LPO problems and has the potential to be used in other nuclear engineering optimization problems. A new modified BA was proposed to solve the optimal management of the multiobjective reconfiguration problem [29]. This algorithm was tested on the 32-bus IEEE radial distribution system and demonstrated its feasibility and effectiveness. The bat-inspired algorithm was combined with differential evolution (BA-DE) into a new hybrid algorithm to solve constrained optimization problems [15]. Comparisons showed that the BA-DE outperformed the most advanced methods in terms of the final solution's quality. A new multiobjective optimization based on the modified BA and Pareto front was developed for solving passive power filter design problems [30]. Local and global search characteristics of BA (EBA) through three different methods were enhanced for solving optimization problems [31]. It was proven that EBA was more effective than other algorithms such as the DE, EA, FA, artificial immune system, and charged system search. BA was used to solve the reliability redundancy allocation problem (RAP) and showed that the BA was competitive with the best known heuristics for redundancy allocation [4]. BA was used to design and optimize the concurrent tolerance in mechanical assemblies [32]. It was found that the BA produced better results than other methods (such as EA and GA) in the initial generations of concurrent tolerance problems. The multidirectional bat algorithm (MDBAT), which hybridized the bat algorithm with the multidirectional search algorithm, was proposed for solving unconstrained global optimization problems [33]. It could accelerate the convergence to the region of the optimal response. The multiobjective bat algorithm was proposed for association rule mining [34]. A novel multiobjective BA was proposed for community detection on dynamic social networks [35]. The algorithm used the mean shift algorithm to generate the new solutions and avoid the random process by defining a new mutation operator. A combination of bat and scaled conjugate gradient algorithms is proposed to improve neural network learning capability [36]. The bat algorithm was first practically implemented in swarm robotics [37]. An autonomous binary version of the bat algorithm was used to solve the patient bed assignment problem [38]. The swarm bat algorithm with improved search was introduced in [39]. The adaptive multiswarm bat algorithm (AMBA) was superior to the others over 20 benchmark functions in a comparison with six algorithms [40]. A hybrid bat algorithm with a genetic crossover operation and smart inertia weight (SGA-BA) was used to solve multilevel image segmentation [41]. A novel bat algorithm with double mutation operators was applied to the low-velocity impact localization problem [42].

3. Standard Bat Algorithm

The standard BA is a bioinspired algorithm based on the echolocation behavior of bats [12]. Bats emit a very loud sound pulse and listen for the echo that bounces back from the surroundings. Bats fly randomly using frequency, velocity, and position to search for prey [43, 44]. In the BA, the frequency, velocity, and position of each bat in the population are updated for further movements. This algorithm is formulated to imitate the behavior of bats finding their prey such that it serves to solve both single objective and multiobjective optimization problems in the continuous solution domain [14]. The implementation of the standard BA is as follows [12–14]:

(1) All bats use echolocation to sense distance, and they also "know" the difference between prey and background barriers in an uncanny way.

(2) With a varied frequency f_i, varied wavelength λ, and loudness A_0, bats fly randomly with velocity v_i at position x_i to search for prey. They can automatically adjust the wavelength (or frequency) according to the distance from the prey.

(3) Although the loudness can vary in different ways, it is assumed that the loudness varies from large A_0 to a minimum value $A_{\min}$.

In the BA, each bat is defined by its position x_i^t, velocity v_i^t, frequency f_i, loudness A_i^t, and the emission pulse rate r_i^t at time t in a D-dimensional search space. The new solution at time t is given by

$$x_i^t = x_i^{t-1} + v_i^t. \tag{1}$$

The velocity is updated by

$$v_i^t = v_i^{t-1} + \left(x_i^{t-1} - x_{gbest}^t\right) f_i, \tag{2}$$

where x_{gbest}^t is the current global best location found by all the bats in the past generations. The pulse frequency f_i is dominated by

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \tag{3}$$

where S is a random number drawn from a uniform distribution. The two parameters $f_{\min}$ and $f_{\max}$ are the minimum and maximum frequency values, respectively.

Whenever a solution is selected as the current optimal solution (x_{gbest}) in the local search areas, the new solution (x_i) is generated by a random walk as follows:

$$x_i = x_{\text{gbest}} + rand * A^t, \tag{4}$$

where *rand* is a random number uniformly distributed within $[-1, 1]$ and controls the direction and power of the random walk. A^t is the average loudness of all the bats.

With the ith bat approaching the prey, its loudness A_i will decrease and its rate r_i of pulse emission will increase. Such a change can be achieved by the following formulas:

$$A_i^{t+1} = \alpha A_i^t, \tag{5}$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)], \tag{6}$$

where γ and γ are constants. For any $0 < \alpha < 1$ and $\gamma > 0$, we have

$$A_i^{t+1} \longrightarrow 0, \quad r_i^t \longrightarrow r_i^0, \text{ as } t \longrightarrow \infty. \tag{7}$$

The pseudocode of the original BA is given in Algorithm 1

4. GIS-Based Niche Hybrid Bat Algorithm (NHBA)

4.1. Brief Description of NHBA. The standard BA has a premature convergence rate and is easily trapped in a local optimum. To overcome these problems, NHBA is proposed by means of the niche technique hybridizing BA. The idea of NHBA is as follows.

First, the Euclidean distance d_{ij} is calculated between the arbitrary two bats Bat_i and Bat_j among the population. If the Euclidean distance d_{ij}, denoted as $\| Bat_i - Bat_j \|$, is less than the niche distance L which is a threshold value and is given previously based on experience, it signifies that the bats B_i and B_j have higher similarity. The fitness values of two individuals are compared. Then, the lower fitness value is given a penalty factor in order to decrease its fitness value. Regarding the two individuals within the niche distance L, the individuals with lower fitness values will become worse

```
Objective function $F(x)$, $x = (x_1,x_2,...,x_d)^T$
Initialize the bat population $x_i$ ($i = 1,2, \ldots, n$) and $v_i$
Define the pulse frequency $f_i$ at $x_i$
Initialize the pulse rates $r_i$ and the loudness $A_i$
While ($t < I_{\max}$) (where $I_{\max}$ is the max number of iterations)
  Generate new solutions by adjusting the frequency, and updating velocities and locations using equations (1)–(3)
  If ($rand1 > r_i$)
    Select a solution among the best solutions
    Generate a local solution around the selected best solution using equation (4)
  End if
  Generate a new solution by flying randomly
  If ($rand2 < A_i$ and $f(x_i) < f(x_{\mathrm{gbest}})$)
    Accept the new solutions
    Increase $r_i$ and reduce $A_i$ using equations (5) and (6)
  End if
  Rank the bats and find the current best $x_{gbest}$
End while
```

Algorithm 1: The pseudocode of the original BA.

and be eliminated by a large probability in the next generations. It means that there exist only elite individuals within the niche distance L. Therefore, the population diversity is maintained. Meanwhile, the two different individuals maintain a certain distance, and all the individuals are scattered over the whole constrained space. Finally, the niche hybrid bat algorithm is implemented in this way.

4.2. GIS-Based NHBA for Optimal Spatial Search. The GIS-based NHBA has the advantage of being applied to optimal spatial search with combined GIS techniques during a sequence of iterations. Optimal spatial search is involved in the resource or location allocation of facilities, which include hospitals, supermarkets, schools, cinemas, and fire stations. Hence, optimal spatial search plays an important role in geographic information science. This paper uses the optimal location allocation of facilities as an example. This type of problem is involved in complex nonlinear optimization scenarios, especially in multiobjective location allocation. If we use the brute-force algorithm, i.e., list all the combinatorial results to obtain the optimal value, the computational volume is so enormous that it is impossible to finish the computing work in an acceptable time. The GIS-based NHBA is an effective algorithm that can easily optimize the complex spatial search. The performance procedure is described as follows.

4.2.1. Define the Objective Function. The objective function is strongly associated with the population density, traffic accessibility, and competitive factors of facilities.

(1) Population Density. Population density is an important factor affecting the spatial distribution of facilities. When a facility is located in a position with a large population density, it will undoubtedly increase the passenger volume. The influence of the facilities on the population rapidly attenuates with the increase in distance. Suppose the central position of the facility is $P(x, y)$, the formula of which has an effect on a certain point $P_j(x'_j, y'_j)$ in the 2-dimensional geographical space is expressed by the following equation:

$$D_j = e^{-k\sqrt{\left(x-x'_j\right)^2+\left(y-y'_j\right)^2}}, \tag{8}$$

where k is the decay coefficient. The greater the value k, the faster the decay is, and vice versa. When the facility scale is large, its influence is also large, and when the value k is relatively low, it means there is a wider scope of influence. In addition, traffic accessibility has a greater impact on the value k. When the transportation is developed, the value k is relatively lower and the influence scope of the facilities is broader. In this paper, we suppose that every facility scale is fixed, which means the value k is constant. The facility with a good position should try to cover the maximum population density, i.e., the formula: max D_j.

(2) Traffic Accessibility. Traffic accessibility is another important factor affecting the spatial distribution of facilities. Convenient traffic can attract a larger passenger volume. As a consequence, the population number will be reduced accordingly with the increasing space distance. Suppose the distance $d(x, y)$ between an arbitrary point $P_j(x'_j, y'_j)$ and the central position of a certain facility $P(x, y)$ is described by the Euclidean distance and expressed by the following equation:

$$d(x, y) = \sqrt{\left(x - x'_j\right)^2 + \left(y - y'_j\right)^2}. \tag{9}$$

Traffic accessibility means that the optimal locations of the facilities meet the minimum sum of the distances between the population and the facilities, i.e., the formula: $\min\sum d_j(x, y)$.

(3) Competitive Factor. Competition exists between the different facilities so that it is necessary to keep a certain distance between the facilities. If the factor of the facility scale is not considered, the proximity principle will be

followed by people. Suppose the minimum distance $d_{j\min}$ between the n facilities $P_1(x_1, y_1)$, $P_2(x_2, y_2)$,. . ., $P_n(x_n, y_n)$ and the jth population cell $P_j(x_j', y_j')$ exists and is given as follows:

$$d_{j\min} = \min d_j\left(x_j, y_j\right) = \min\sqrt{\left(x_i - x_j'\right)^2 + \left(y_i - y_j'\right)^2}, \tag{10}$$

where $i = 1, 2,. . ., n$; $j = 1, 2, . . ., h$. The parameters n and h are the number of facilities and the number of population cells, respectively.

4.2.2. Objective Function. According to the above definitions, the problem should be how to locate the facilities in a certain place when the number of facilities is constant. First, the objective function is defined to solve the problem. The influence of the facilities will gradually attenuate with the distance. Thus, the objective function based on the vector data is calculated as follows:

$$F = \frac{C\sum_{i=1}^{n}\sum_{j=1}^{h} D_j A_j e^{-kd_{ij}}}{\sum_{j=1}^{h} D_j A_j d_{j\min}}, \tag{11}$$

where F is the fitness value of the objective function; C is a constant; D_j is the population density of the jth population cell; A_j is the area of the jth population cell, where $j = 1,2,. . .,h$; and k is the decay parameter. The parameter d_{ij} is equal to $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, where d_{ij} is the distance between point (x_i, y_i) and point (x_j, y_j). The point (x_i, y_i) represents the coordinates of the ith facility, and the point (x_j, y_j) represents the coordinates of the jth population cell.

The combinatorial optimization problem can be formulated as computing the maximum value. Therefore, the purpose of the research is to obtain the coordinates of facilities $P_g(x_{gbest}, y_{gbest})$ under the constraint condition of the maximization of F.

4.2.3. Introduction to NHBA. The NHBA treats each solution as a bat searching in D-dimensional hyperspace. Different from other optimal problems, the optimal spatial search is in a 2-dimensional geographic space in which each point includes X and Y coordinates. The concept of geographic multidimensional space is different from that of D-dimensional hyperspace. D is equal to $2n$, where n is the number of facilities. Consequently, the position vector of the ith bat is $B_i\,(x_{i1}, y_{i1}, x_{i2}, y_{i2}, . . ., x_{in}, y_{in})$. Each bat flies over the search space, and its velocity vector is $V_I\,(v_{xi1}, v_{yi1}, v_{xi2}, v_{yi2}, \ldots, v_{xin}, v_{yin})$. The bats can adjust their positions and velocities according to the current optimal value $B(j).Best$ and global optimal value B_{gbest}. The updating formula of the position is expressed by

$$\begin{cases} x_i\,(t+1) = x_i\,(t) + v_{xi}\,(t+1), \\ y_i\,(t+1) = y_i\,(t) + v_{yi}\,(t+1). \end{cases} \tag{12}$$

The velocity is updated by

$$\begin{cases} v_{xi}\,(t+1) = v_{xi}\,(t) + \left(x_i\,(t) - x_{gbest}\,(t)\right) f_{xi}, \\ v_{yi}\,(t+1) = v_{yi}\,(t) + \left(y_i\,(t) - y_{gbest}\,(t)\right) f_{yi}, \end{cases} \tag{13}$$

where t is the iteration. The parameters $x_{gbest}\,(t)$ and $y_{gbest}\,(t)$ are the global optimal positions of X and Y axis directions found by all the bats in the tth iteration. The frequency f is dominated by

$$\begin{cases} f_{xi} = f_{x\min} + \left(f_{x\max} - f_{x\min}\right)\beta_{xi}, \\ f_{yi} = f_{y\min} + \left(f_{y\max} - f_{y\min}\right)\beta_{yi}, \end{cases} \tag{14}$$

where β_{xi} and β_{yi} are random vectors subjected to a uniform distribution in the range of [0, 1]. Typically, $f_{xmin} = f_{ymin} = 0$, $f_{xmax} = f_{ymax} = 100$, and the initial frequencies of each bat along the X and Y axis directions are uniformly distributed random numbers in the range of $[f_{xmin}, f_{xmax}]$ and $[f_{ymin}, f_{ymax}]$, respectively:

$$\begin{cases} x_{inew}\,(t+1) = x_{iold}\,(t) + \varepsilon_{xi}\,(t) \cdot A_{xi}\,(t), \\ y_{inew}\,(t+1) = y_{iold}\,(t) + \varepsilon_{yi}\,(t) \cdot A_{yi}\,(t), \end{cases} \tag{15}$$

wherein $\varepsilon_{xi}\,(t)$ and $\varepsilon_{yi}\,(t)$ are random numbers in the range of $[-1, 1]$ and $A_{xi}\,(t)$ and $A_{yi}\,(t)$ are the average loudness of all the bats along the X and Y axis directions in the tth iteration:

$$\begin{cases} A_{xi}\,(t+1) = \alpha \cdot A_{xi}\,(t), \\ A_{yi}\,(t+1) = \alpha \cdot A_{yi}\,(t). \end{cases} \tag{16}$$

The parameter α is the adjustment coefficient and takes the appropriate value according to the specific problem:

$$\begin{cases} A_{xi}\,(t) = \sum\limits_{j=1,3,5}^{2n-1} \dfrac{r_{xij}\,(t)}{n}, \\ \\ A_{yi}\,(t) = \sum\limits_{j=2,4,6}^{2n} \dfrac{r_{xij}\,(t)}{n}, \end{cases} \tag{17}$$

$$\begin{cases} r_{xi}\,(t) = r_{xi}\,(0) \cdot \left(1 - e^{-\gamma t}\right), \\ r_{yi}\,(t) = r_{yi}\,(0) \cdot \left(1 - e^{-\gamma t}\right). \end{cases} \tag{18}$$

The parameters $r_{xi}\,(0)$ and $r_{yi}\,(0)$ are the initial pulse rates, $r_{xi}\,(t)$ and $r_{yi}\,(t)$ are the tth iteration of the ith bat, and γ is the adjustment coefficient. Because the x and y coordinates pair up, $j = 1, 3, 5,. . ., 2n{-}1$ and $j = 2, 4, 6, \ldots, 2n$ along the X- and Y-axis directions, respectively.

4.2.4. Implementing the Procedure of NHBA for Optimal Spatial Search. Based on the above concepts, the implementation procedure of NHBA is described as follows:

(1) Define the objective function $F(B_i)$ based on equation (11).

(2) Initialize the parameters for each bat: the population size M, position x, velocity v, loudness A, and pulse rate r.

(3) Set the arithmometer that is used to count the number of bats. Generate M bats from the initial population with random positions and velocities, and calculate the fitness value $F(B_i)$ of each bat of

the population based on the objective function in step 1.

(4) Evaluate each bat and find the current optimal bat in each population and save it as *B*(*j*).*Best.*

(5) Begin to iterate from each bat in the population.

(5.1) Calculate the frequency, velocity, and position for the *i*th bat based on the equations (12)–(14).

(5.2) Judge whether the positions x_i and y_i of all the bats exceed the boundaries. If so, let x_i and y_i be equal to the critical values of boundaries.

(5.3) If $rand > r_{xi}$ or $rand > r_{yi}$, generate a new solution for each bat based on equation (15). A random walk procedure has been used to generate a new solution for each bat to improve the variability of the possible solution, where the parameter *rand* is a random number and $rand \in$ [-1, 1].

(5.4) Adjust the loudness A and pulse rates r of all the bats in the tth iteration. At each iteration of the algorithm, the loudness $A_x(t)$ and $A_y(t)$ and the pulse rates $r_{xi}(t)$ and $r_{yi}(t)$ are updated based on the equations (16)–(18) if the new solutions are improved. It means that these bats are moving toward the optimal solution.

(5.5) A solution is accepted if a number drawn at random is less than the loudness Ai and meets the relation F(Bi) >B$_{gbest}$, where B$_{gbest}$ is the global optimum value.

(5.6) Update the frequency, velocity, position, loudness, pulse rate, and fitness values of each bat because the values of the parameters change.

(6) Calculate the Euclidean distance d_{ij} between two arbitrary bats B_i and B_j in the population, where $d_{ij} = \|B_i - B_j\| = \sqrt{\sum_{k=1}^{n}[(x_{ik} - x_{jk})^2 + (y_{ik} - y_{jk})^2]}$.

(7) Implement the elimination algorithm of the niche. When the relation $d_{ij} < L$ is met, compare the fitness values of two bats B_i and B_j. L is the niche distance. It is an empirical value and is given based on experience in the following experiment (see Tables 1 and 2). Impose the stronger penalty factor on the bat with the lower fitness value, i.e., $\min(F(B_i), F(B_j)) = Penalty$, so that the diversity of the population is maintained. The *Penalty* is a very small number. For example, we let $Penalty = 10^{-6}$. By doing so, it can help the population improve its average fitness value and accelerate the speed of obtaining the optimal value.

(8) The new population is sorted in descending order by the new fitness values. The bats, of which the fitness values are less than the elimination factor *elm*, are divided into the population *V*. The others are divided into the population *W* and memorized. The bats of population *V* will be eliminated and reinitialized.

Table 1: List of parameters when $n = 1$.

Parameters	Meaning	Selected values
M	The number of bats	80
C	Constant	100,000
K	Decay coefficient	0.0004
c_1, c_2	Constant	2
$E_{\text{ min}}$	Minimum error	10^{-6}
α	The adjustment coefficient	0.90
γ	Constant, $\gamma > 0$	0.96
L	The niche distance	500

Table 2: List of parameters when $n > 1$.

Parameters	Meaning	Values
M	The number of bats	108
C	Constant	100,000
K	Decay coefficient	0.0004
$E_{\min}$	Minimum error	10^{-6}
α	The adjustment coefficient	0.86
γ	Constant, γ>0	0.95
L	The niche distance	2000

(9) Judge whether it meets the termination condition. If it does not meet the termination condition, update the arithmometer $t = t + 1$. The former *W* individuals produced in step 8 are treated as the new population of the next iteration. Then, go to step 5. If the termination condition is met, i.e., $|F_{t+1} - F_t| < \varepsilon$ (a critical value) or the maximum iterations are exceeded, end the algorithm.

(10) Obtain the best solution, including the best positions and best fitness values.

(11) Memorize and output the results.

5. Experiment and Discussion

5.1. Data and Methods. The spatial search needs to combine the location data with the attribute data. Regarding the problems of locating the facilities in a certain place, GIS can provide robust support. Some important information, which includes the centroid coordinates (x_i, y_i) and area of each population cell and the width and height of the search space, needs to be extracted from the map of the population density distribution by means of GIS. Python, a common language tool, is used in this research. We can import and manipulate the input data and thus present the generated map.

The digitalized population density map is a polygon vector map of Fangcun District, Guangzhou City, China. Each polygon represents a population cell, which is the minimal unit of the geographic area. The fields should include *OID* and *Population* in the attribute database of GIS. The field *OID* represents the object identity code, which is unique. The field *Population* represents the population density of each population cell, which can be extracted from the attribute database of GIS. Compared with the raster data, the precision of the vector data we use in the paper is high.

The map of the population density in Fangcun District, Guangzhou City, China, is shown in Figure 1.

5.2. Case for Single Facility ($n = 1$). This is the simplest case when $n = 1$ because there is no competition from other facilities. The parameters are listed in Table 1.

The optimal value is $F_{\max} = 22.0968$ after 6 iterations based on NHBA. The results are shown in Figure 2.

We compare NHBA with BA, GA, FA, PSO, and the brute-force algorithm under the same parameters (see Table 1) by our programming in C language. The comparison results are shown in Tables 3 and 4.

The optimal value of BA is $F_{\max} = 22.0967$ after 7 iterations. NHBA has fewer iterations and a higher precision than BA under the same parameters. GA achieves a similar optimal value with the same iterations compared with BA. The optimal value of PSO is $F_{\max} = 22.0968$ after 24 iterations. The results are similar among PSO, BA, and NHBA. However, PSO needs more iterations. FA obtains a smaller optimal value with more iterations. It means that FA is the worst algorithm and NHBA is the best algorithm among the five algorithms.

The brute-force algorithm is used in order to validate the results. The time complexities of the algorithm and the accuracies of the solutions are closely related to the spatial sampling interval. The results of the fitness values, time complexities, and x and y coordinates are shown in Table 4.

The smaller the spatial sampling interval is, the greater the time complexities are, and the higher the accuracies of the solutions are. When the spatial sampling interval is 1×1 meters, the fitness value is 22.096775, which is roughly similar to most algorithms. However, the time complexity of the brute-force algorithm is $O\ (8.6 \times 10^7)$. BA needs 7 iterations, and the time complexity is $O\ (1.22 \times 10^6)$. NHBA only needs 6 iterations, and the time complexity is $O\ (1.05 \times 10^6)$. Consequently, NHBA achieves a faster convergence speed.

The 3D perspective map regarding the fitness values and geographical positions in Fangcun District is shown in Figure 3.

The map is obtained by means of developing techniques based on Python and R languages. First, evenly generate $100*100$ original sample points in the 2D space within the Fangcun District boundary. Then, calculate the fitness values as z_i coordinates corresponding to every point (x_i, y_i) within the boundary. Finally, create the 3D perspective map based on every point (x_i, y_i, z_i) in R language.

In Figure 3, the x and y coordinates represent the position of a facility; the z coordinate represents its fitness value. We can roughly judge the optimal position in which a facility is located based on Figure 3.

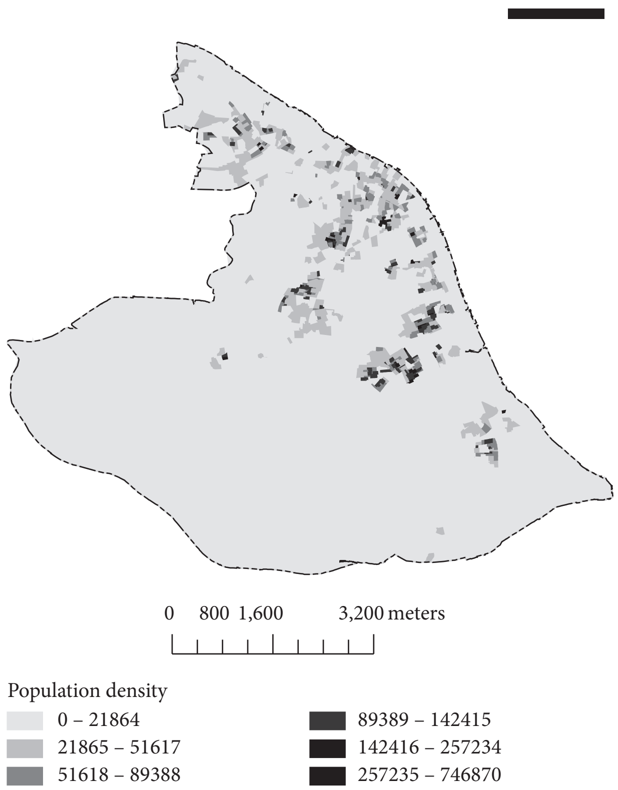

Figure 1: Map of population density in Fangcun District.

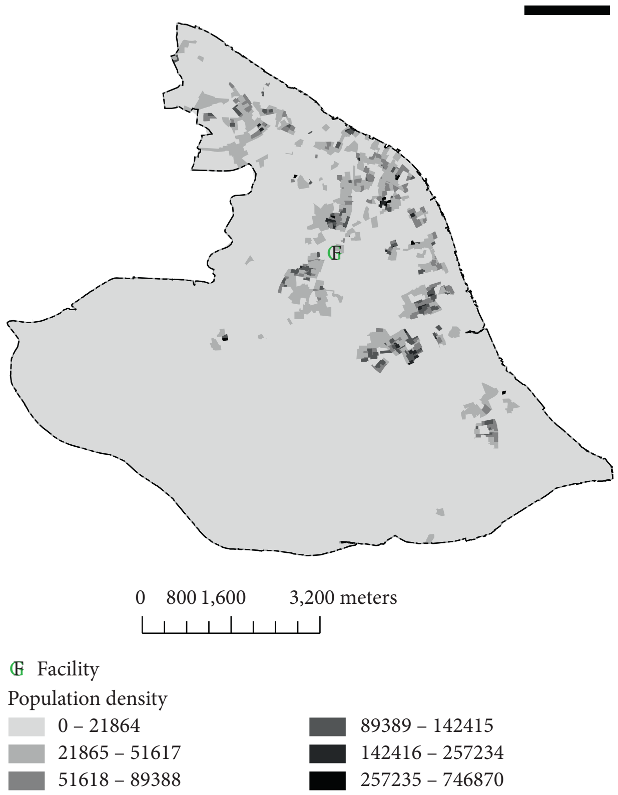

Figure 2: Map of the population density and facility distribution ($n = 1$) based on NHBA in Fangcun District.

5.3. Case for Two and More Facilities ($n > 1$). When the number of facilities n is more than 1, it is considered a combinatorial optimization problem, which is computationally difficult. For example, if the brute-force algorithm is

Table 3: The comparison results of NHBA with BA, GA, FA, and PSO.

Algorithms	Optimal values	Iterations	Time complexities
BA	22.0967	7	1.22×10^6
GA	22.0965	7	1.22×10^6
FA	21.0478	95	1.66×10^7
PSO	22.0968	24	4.20×10^6
NHBA	22.0968	6	1.05×10^6

Table 4: Results of the brute-force algorithm based on the spatial sampling interval.

Spatial sampling interval	Fitness values	X	Y	Time complexities
100×100	21.188921	34579.5	24920.4	8712
10×10	22.095361	34079.5	25020.4	863625
1×1	22.096775	34066.5	25038.4	8.6×10^7

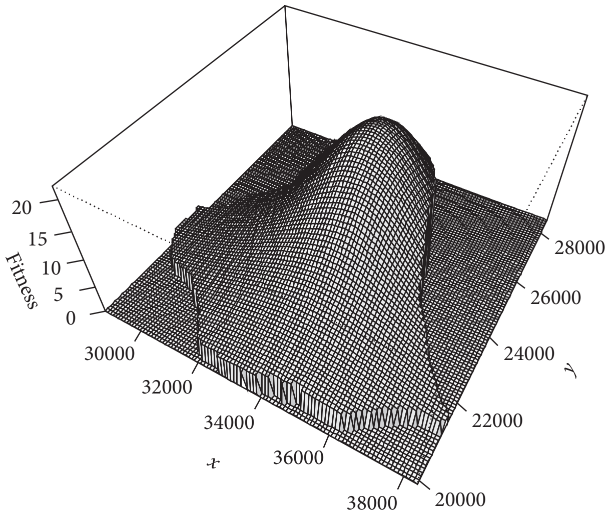

Figure 3: The 3D perspective map of the fitness values and geographical positions in Fangcun District.

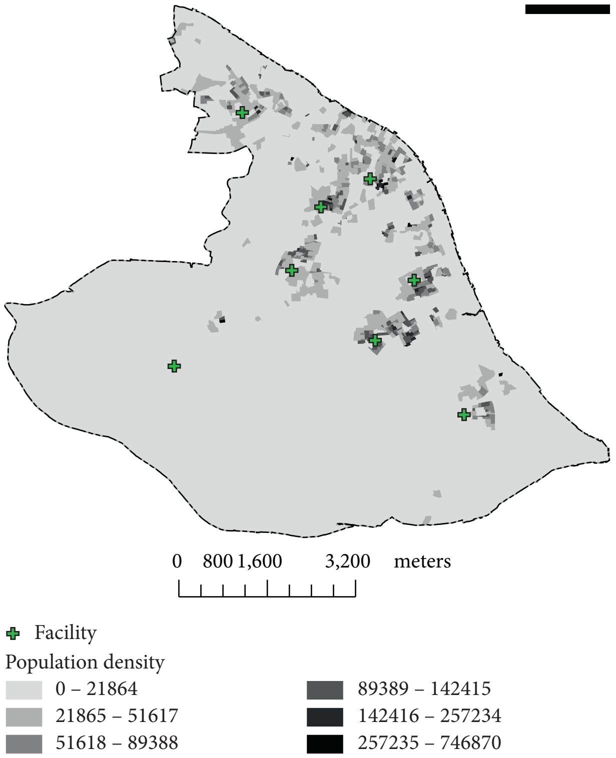

Figure 4: Map of the population density and facilities distribution ($n = 8$) based on NHBA in Fangcun District.

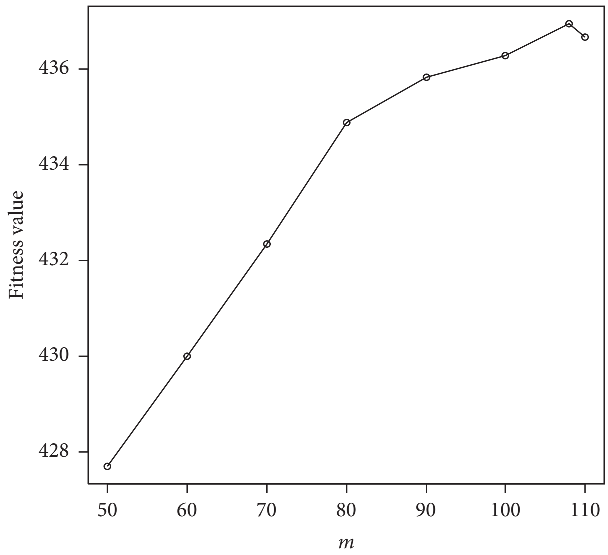

Figure 5: The relationship between the parameter m and the fitness values.

used, the time complexity is $O\ (7.65 \times 10^{58})$ when the spatial sampling interval is 1 meter and $n = 8$. Because the computing time is unacceptable, the brute-force algorithm has to be abandoned.

The parameters are listed in Table 2. The optimal value $F_{\text{max}} = 436.94$ after 13 iterations ($GEN_{\text{NHBA}} = 13$) based on NHBA. The results are shown in Figure 4.

In general, the parameter m, namely, the population size or number of bats, is given based on the researcher's experience. In this paper, the parameter m is obtained by conducting experiments.

In Figure 5, the larger the parameter m is, the larger the fitness value is (that means that the results are better) when m is lower than 108. However, the fitness value becomes lower when m is equal to 110. In addition, the computation time will increase with the increasing m value. As a result, the parameter m is 108 when the number of facilities is 8. The adjustment coefficients are $\alpha = 0.86$ and $\gamma = 0.95$ in this experiment. The comparative results among the BA, GA, FA, PSO, and NHBA algorithms based on the same parameters are as follows.

The fitness values are calculated by the objective function $F(B_i)$ based on equation (1). The larger the fitness values are,

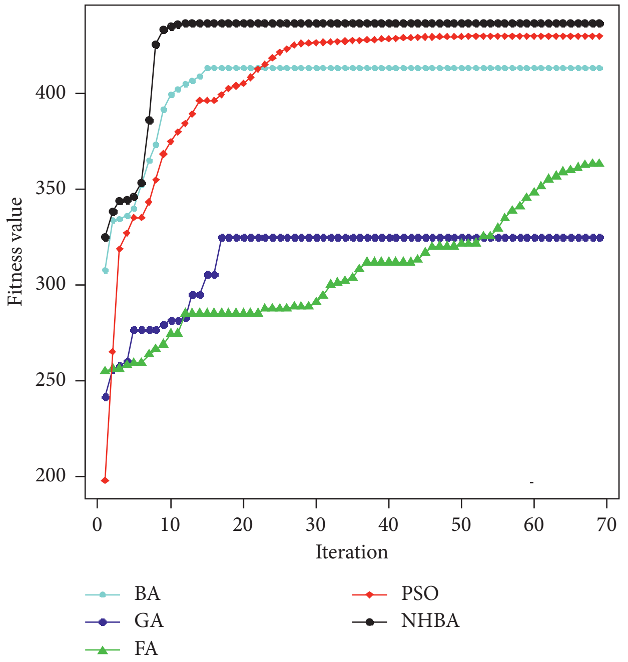

Figure 6: The comparative results among the BA, GA, FA, PSO, and NHBA algorithms with the fitness values and iterations.

the better the results are. In Figure 6, the fitness values of BA, GA, FA, PSO, and NHBA increase with the iterations. The fitness values of NHBA are larger than those of the other algorithms under the same iterations. The fitness values of NHBA tend to be stable under 13 iterations ($GEN_{\mathrm{NHBA}} = 13$), while BA needs 15 iterations, GA needs 17 iterations, FA needs 68 iterations, and PSO needs 54 iterations. The maximum fitness values of BA, GA, FA, PSO, and NHBA are 413.42, 324.80, 363.32, 430.85, and 436.94, respectively. It is easier for NHBA to obtain the optimal values than it is for the other algorithms. This finding means that the NHBA approach has the ability to converge to optimal solutions more quickly than other algorithms.

The time complexities are defined as $O\,(GEN^{*} m^{*} 2n^{*} h)$. Here, GEN is the number of iterations, m is the population size ($m = 108$), n is the number of facilities ($n = 8$), and h is the number of population cells ($h = 1093$). Consequently, the time complexities of BA, GA, FA, PSO, and NHBA are approximately $O\ (2.83 \times 10^{7})$, $O\ (3.21 \times 10^{7})$, $O\ (1.28 \times 10^{8})$, $O\ (1.02 \times 10^{8})$, and $O\ (2.46 \times 10^{7})$ in this experiment, respectively (Table 5). As mentioned above, the time complexity of the brute-force algorithm is $O\ (7.65 \times 10^{58})$ under the same condition. In conclusion, the convergence of NHBA is faster than that of the other algorithms. Furthermore, the computation time of BA, GA, FA, PSO, or NHBA is significantly less than that of the brute-force algorithm.

5.4. Experiment Based on the Expanded Shaffer Function. The NHBA has random elements, and the statistics of only one experimental result are not convincing. The expanded Shaffer function (equations (19) and (20)) is used to test the above algorithms:

Table 5: Comparison results of BA, GA, FA, PSO, and NHBA.

Methods	Fitness values	Iterations	Time complexities
BA	413.42	15	2.83×10^{7}
GA	324.80	17	3.21×10^{7}
FA	363.32	68	1.28×10^{8}
PSO	430.85	54	1.02×10^{8}
NHBA	436.94	13	2.46×10^{7}

Table 6: Comparison results of NHBA and other algorithms based on the expanded Shaffer function after 50 runs ($D = 50$).

Methods	Means	Best optimal values	Worst optimal values	Variances
BA	21.47	19.58	22.79	460.91
GA	23.78	23.22	24.38	565.49
FA	15.28	13.25	16.98	233.50
PSO	17.59	14.60	20.06	1.30
NHBA	13.14	9.20	18.03	172.78

$$g\,(x, y) = 0.5 + \frac{\sin^{2}\sqrt{x^{2} + y^{2}} - 0.5}{\left[1.0 + 0.001\left(x^{2} + y^{2}\right)\right]^{2}}\ st\ -100 < x, \quad y < 100, \tag{19}$$

$$\begin{aligned} f\,(x) = {} & g\left(x_{1}, x_{2}\right) + g\left(x_{2}, x_{3}\right) + g\left(x_{3}, x_{4}\right) + \ldots \\ & + g\left(x_{D-1}, x_{D}\right) + g\left(x_{D}, x_{1}\right). \end{aligned} \tag{20}$$

When $x(x_{1}, x_{2}, \ldots, x_{D})$ locates in the position $(0, 0, \ldots, 0)$, the minimum value of the expanded Shaffer function is equal to 0. After every algorithm is run 50 times in D-dimensional space ($D = 50$), the statistical means, the best optimal values, the worst optimal values, and the variances of experimental results are obtained and listed in Table 6.

In Table 6, from the means, we can see that the mean of NHBA is 13.14, which is the smallest. The second smallest mean is achieved by FA (15.28), and the others in ascending order are as follows: PSO (17.59), BA (21.47), and GA (23.78). The order of the best optimal values is the same as that of the means. However, the smallest value of the worst optimal values is obtained by FA instead of NHBA. The smallest variance is obtained by PSO (1.30). It means that the numerical change range of PSO is the smallest. Because the means can better reflect the overall level, it means that NHBA surpasses other methods and GA is worse than other algorithms.

6. Conclusions

Complicated combinatorial optimization problems are involved in the optimal spatial search and are difficult to be solved by traditional algorithms such as the brute-force algorithm. In this paper, a GIS-based NHBA is proposed for solving the optimal spatial search. It is characterized as follows:

(1) The GIS-based NHBA is able to maintain the population diversity, enhance the exploration of the new search domain, and overcome the deficiency that

occurs in the premature convergence to only one solution of optimal spatial search.

(2) The spatial search occurs in multidimensional geographical space. Optimal spatial search involves more complicated optimization problems. The GIS-based NHBA provides robust support for this work.

(3) The parameter population size m is traditionally obtained by the researcher's experience. It is derived by the experiments conducted in this paper in order to improve the precision of the results.

(4) The BA, GA, FA, PSO, and NHBA outperform the brute-force algorithm in terms of computational efficiency.

(5) The optimal value is easier to obtain by employing NHBA than it is by using BA, GA, FA, and PSO. Furthermore, the precision of NHBA is higher and the convergence of NHBA is faster than that of BA, GA, FA, and PSO under the same conditions.

Data Availability

The data used to support the findings of this study were supplied under license and so cannot be made freely available. Requests for access to these data should be made to Guoming Du via eesdgm@mail.sysu.edu.cn.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Program on Key Research Projects of China (no. 2017YFC1502706). It was also partially and financially supported by the Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai) (no. 99147-42080011) and the Hundred Talents Program of Sun Yat-Sen University (37000-18841201).

References

[1] T. A. Arentze, A. W. J. Borgers, and H. J. P. Timmermans, "An efficient search strategy for site-selection decisions in an expert system," *Geographical Analysis*, vol. 28, no. 2, pp. 126–146, 1996.

[2] J. C. J. H. Aerts and G. B. M. Heuvelink, "Using simulated annealing for resource allocation," *International Journal of Geographical Information Science*, vol. 16, no. 6, pp. 571–587, 2002.

[3] X. S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.

[4] T. P. Talafuse and E. A. Pohl, "A bat algorithm for the redundancy allocation problem," *Engineering Optimization*, vol. 48, no. 5, pp. 900–910, 2016.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Ardor, Michigan, 1975.

[6] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, "Ant system for job-shop scheduling," *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 34, no. 1, pp. 39–54, 1994.

[7] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, Australia, December 1995.

[8] X. L. Li, Z. J. Shao, and J. X. Qian, "s," *System Engineering Theory and Practice*, vol. 22, no. 11, pp. 32–38, 2002, in Chinese.

[9] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings Of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214, IEEE, Coimbatore, India, December 2009.

[10] R. Q. Zhao and W. S. Tang, "Monkey algorithm for global numerical optimization," *Journal of Uncertain Systems*, vol. 2, no. 3, pp. 165–176, 2008.

[11] X. S. Yang, *Nature-inspired Metaheuristic Algorithms*, pp. 83–96, Luniver Press, London, UK, 2008.

[12] X. S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Vol. 284 of Studies in Computational Intelligence, pp. 65–74, Springer, Berlin, Heidelberg, 2010.

[13] M. Zineddine, "Optimizing security and quality of service in a real-time operating system using multi-objective Bat algorithm," *Future Generation Computer Systems*, vol. 87, pp. 102–114, 2018.

[14] X. S. Yang, "Bat algorithm for multi-objective optimisation," *International Journal of Bio-Inspired Computation*, vol. 3, no. 5, pp. 267–274, 2011.

[15] S. Y. Pei, A. Ouyang, and L. Tong, "A hybrid algorithm based on bat-inspired algorithm and differential evolution for constrained optimization problems," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 4, Article ID 1559007, 2015.

[16] L. Wei and M. Zhao, "A niche hybrid genetic algorithm for global optimization of continuous multimodal functions," *Applied Mathematics and Computation*, vol. 160, no. 3, pp. 649–661, 2005.

[17] X. M. Zhang and L. R. Wang, "Two-parameter inversion of fluid-saturated porous medium with niche ant colony algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 164932, 6 pages, 2014.

[18] D. W. Coit and A. E. Smith, "Reliability optimization of series-parallel systems using a genetic algorithm," *IEEE Transactions on Reliability*, vol. 45, no. 2, pp. 254–260, 1996.

[19] V. Hajipour, S. H. A. Rahmati, S. H. R. Pasandideh, and S. T. A. Niaki, "A multi-objective harmony search algorithm to optimize multi-server location-allocation problem in congested systems," *Computers & Industrial Engineering*, vol. 72, pp. 187–197, 2014.

[20] S. M. Mousavi, N. Alikar, S. T. A. Niaki, and A. Bahreininejad, "Optimizing a location allocation-inventory problem in a two-echelon supply chain network: a modified fruit fly optimization algorithm," *Computers & Industrial Engineering*, vol. 87, pp. 543–560, 2015.

[21] J.-P. Arnaout, G. Arnaout, and J. E. Khoury, "Simulation and optimization of ant colony optimization algorithm for the stochastic uncapacitated location-allocation problem," *Journal of Industrial and Management Optimization*, vol. 12, no. 4, pp. 1215–1225, 2016.

[22] M. Alinaghian and A. Goli, "Location, allocation and routing of temporary health centers in rural areas in crisis, solved by improved harmony search algorithm," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 894–913, 2017.

[23] Y. Lin, H. Jia, Y. Yang, G. Tian, F. Tao, and L. Ling, "An improved artificial bee colony for facility location allocation problem of end-of-life vehicles recovery network," *Journal of Cleaner Production*, vol. 205, pp. 134–144, 2018.

[24] E. B. Tirkolaee, J. Mahmoodkhani, M. R. Bourani, and R. Tavakkoli-Moghaddam, "A self-learning particle swarm optimization for robust multi-echelon capacitated location-allocation-inventory problem," *Journal of Advanced Manufacturing Systems*, vol. 18, no. 04, pp. 677–694, 2019.

[25] A. H. Gandomi, X. S. Yang, A. H. Alavi, and S. Talatahari, *Bat Algorithm for Constrained Optimization Tasks*, Springer-Verlag, London, UK, 2012.

[26] S. K. Saha, R. Kar, D. Mandal, S. P. Ghoshal, and V. Mukherjee, "A new design method using opposition-based bat algorithm for IIR system identification problem," *International Journal of Bio-Inspired Computation*, vol. 5, no. 2, pp. 99–132, 2013.

[27] C. Sur and A. Shukla, "Adaptive & discrete real bat algorithms for route search optimization of graph based road network," in *Proceedings of the 2013 International Conference on Machine Intelligence and Research Advancement (ICMIRA 2013)*, pp. 120–124, IEEE, Katra, India, December 2013.

[28] S. Kashi, A. Minuchehr, N. Poursalehi, and A. Zolfaghari, "Bat algorithm for the fuel arrangement optimization of reactor core," *Annals of Nuclear Energy*, vol. 64, pp. 144–151, 2014.

[29] M. Golmaryami, S. Saleh, A. B. Ardekani, and F. Kavousi-Fard, "A new modified bat algorithm to solve optimal management of multi-objective reconfiguration problem," *Journal of Intelligent & Fuzzy Systems*, vol. 27, no. 3, pp. 1567–1573, 2014.

[30] N.-C. Yang and M.-D. Le, "Optimal design of passive power filters based on multi-objective bat algorithm and pareto front," *Applied Soft Computing*, vol. 35, pp. 257–266, 2015.

[31] S. Yilmaz and E. U. Kucuksille, "A new modification approach on bat algorithm for solving optimization problems," *Applied Soft Computing*, vol. 28, pp. 259–275, 2015.

[32] L. R. Kumar, K. P. Padmanaban, S. G. Kumar, and C. Balamurugan, "Design and optimization of concurrent tolerance in mechanical assemblies using bat algorithm," *Journal of Mechanical Science and Technology*, vol. 30, no. 6, pp. 2601–2614, 2016.

[33] M. A. Tawhid and A. F. Ali, "Multi-directional bat algorithm for solving unconstrained optimization problems," *Opsearch*, vol. 54, no. 4, pp. 684–705, 2017.

[34] K. E. Heraguemi, N. Kamel, and H. Drias, "Multi-objective bat algorithm for mining numerical association rules," *International Journal of Bio-Inspired Computation*, vol. 11, no. 4, pp. 239–248, 2018.

[35] I. Messaoudi and N. Kamel, "A multi-objective bat algorithm for community detection on dynamic social networks," *Applied Intelligence*, vol. 49, no. 6, pp. 2119–2136, 2019.

[36] P. M. R. Bento, J. A. N. Pombo, M. R. A. Calado, and S. J. P. S. Mariano, "Optimization of neural network with wavelet transform and improved data selection using bat algorithm for short-term load forecasting," *Neurocomputing*, vol. 358, pp. 53–71, 2019.

[37] P. Suárez, A. Iglesias, and A. Gálvez, "Make robots be bats: specializing robotic swarms to the Bat algorithm," *Swarm and Evolutionary Computation*, vol. 44, pp. 113–129, 2019.

[38] C. Taramasco, R. Olivares, R. Munoz, R. Soto, M. Villar, and V. H. C. Albuquerque, "The patient bed assignment problem solved by autonomous bat algorithm," *Applied Soft Computing*, vol. 81, 2019.

[39] R. Chaudhary and H. Banati, "Swarm bat algorithm with improved search (SBAIS)," *Soft Computing*, vol. 23, no. 22, pp. 11461–11491, 2019.

[40] R. Chaudhary and H. Banati, "Adaptive multi-swarm bat algorithm (AMBA)," in *Advances in Intelligent Systems and Computing*, K. Das, J. Bansal, K. Deep, A. Nagar, P. Pathipooranam, and R. Naidu, Eds., vol. 1048, pp. 805–821, 2020.

[41] X. Yue and H. Zhang, "Modified hybrid bat algorithm with genetic crossover operation and smart inertia weight for multilevel image segmentation," *Applied Soft Computing*, vol. 90, 2020.

[42] Q. Liu, J. Li, L. Wu, F. Wang, and W. Xiao, "A novel bat algorithm with double mutation operators and its application to low-velocity impact localization problem," *Engineering Applications of Artificial Intelligence*, vol. 90, 2020.

[43] J. D. Altringham, *Bats: Biology and Behavior*, Oxford University Press, Oxford, UK, 1996.

[44] P. Richardson, *Bats*, Natural History Museum, London, UK, 2008.