https://dx.doi.org/10.1016/j.engappai.2023.106331

Backtracking Search Algorithm with Dynamic Population for Energy Optimization of a UAV-Assisted IoT Data Collection System

Yiying Zhang[a], Chao Huang[b], Hailong Huang[c,*]

[a] School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China
[b] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China
[c] Department of Aerotical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China

Abstract In recent years, collecting data from IoT devices by unmanned aerial vehicles (UAVs) has become a hot and interesting research topic. This paper focuses on using metaheuristic algorithm for the energy consumption of a UAV-based IoT data collection system. The energy consumption problem of this system can be converted into an optimization problem. To solve this optimization problem, this paper presents an improved version of backtracking search algorithm called backtracking search algorithm with a dynamic population (BSADP), which can determine the number and locations of the stop points of the UAV simultaneously. Further, BSADP has a simple framework, which consists of the proposed enhanced backtracking search algorithm (EBSA) and the designed population adjustment mechanism with opposition-based learning (PAMOBL). In the search process, the whole population is regarded as the entire deployment of the UAV. BSADP firstly generates the trail deployment of the UAV by EBSA and then the next generation deployment of the UAV is produced based on the trail deployment and PAMOBL. The performance of BSADP is investigated by two different energy consumption formulations. Experimental results support the superiority of BSADP in optimizing the flight planning of the UAV collecting data from IoT devices and prove the application value of BSADP in the real scenarios.

Keywords unmanned aerial vehicles; energy consumption; Internet of Things; backtracking search algorithm

1 Introduction

In the past several years, smart city has been attracting great attention. The goal of smart city is aimed at improving the quality of life of citizens by making use of modern technology (Belhadi et al., 2020; Duan et al., 2019). Here, it is worth mentioning that Internet of Things (IoT) plays a very important role in the process of building smart city (Qureshi et al., 2020). Further, the core goal of IoT is to maximize the communication of hardware devices with the physical world and to convert automatically the data collected from these devices into some useful information without human assistance (Hammi et al., 2020). An IoT device usually consists of the following three components, i.e. battery-powered embedded sensors, actuators, and communication systems (Kaur & Sood, 2017). When applied to collect and transmit data, the IoT device will consume its energy. To increase the accuracy of the obtained information, more data needs to be collected and transmitted, which will consume more energy. Note that, the energy powered by the battery of the IoT device is limited (Khan et al., 2022). Thus, a valuable problem for an IoT system is how to efficiently collect data from IoT devices.

At present, collecting data from IoT devices by unmanned aerial vehicles (UAVs) has become a hot research topic due to the obvious advantages of UAV-based aerial stations over terrestrial base stations. The advantages mainly refer to the following three aspects (Mozaffari et al., 2017): (1) given the flight altitude of the UAV, the line-of-sight communication links between the UAV and the potential IoT devices can be established, which can potentially improve the quality of wireless communication; (2) the UAV can move flexibly toward the potential IoT devices, which can enable a short-range communication with low energy consumption; (3) the UAV can offer the essential services to the potential IoT devices under some unexpected events, such as updating the embedded programme in the IoT devices. In view of these advantages, UAV-based aerial stations can offer a low cost, energy-efficient and flexible solution to collect data from IoT devices.

In terms of energy consumption, when the UAV is regarded as an IoT data collection platform, the energy consumption of the system involves that of the UAV and that of the IoT devices. As mentioned earlier, the energy consumption of IoT devices is mainly related to data collection and data transmission. For the UAV, its energy consumption is mainly associated with the deployment, i.e. the locations and the number of stop points. A lot of work has been done for the deployment of

* Corresponding author
E-mail address: zhangyiying@ujs.edu.cn (Y. Zhang), hchao.huang@polyu.edu.hk (C. Huang), hailong.huang@polyu.edu.hk (H. Huang)

UAVs. Wang et al. (Wang et al., 2019) investigated the joint optimization of deployment and task scheduling of a UAV by the evolutionary algorithm. Li et al. (Li et al., 2019) built a new UAV energy consumption model, where the particle swarm optimizer was used to find the near-optimal deployment of the UAV subject to a seamless coverage constraint. Huang et al. (Huang & Savkin, 2019) optimized the deployment of a UAV team with the aim of reducing interference effects and maximizing coverage. Chou et al. (Chou et al., 2020) studied the 3D deployment problem for a swarm of UAVs with the goal of maximizing the total data amount transmitted by UAVs. Jiang et al. (Jiang et al., 2019) focused on the joint optimization of the deployment of a UAV and the time allocation with the goal of making the uplink sum achievable rate for all users as large as possible. However, these references only pay attention to optimizing the locations of stop points and don't discuss the influence of the number of the stop points. Note that, the number of the stop points is closely associated with the energy consumption of a UAV (Huang et al., 2019). To deal with this problem, in the latest study, Huang et al. (Huang et al., 2019) reported differential evolution with a variable population size (DEVIPS) to optimize the locations and the number of the stop points simultaneously in a UAV-assisted IoT data collection system. Although this research is a quite novel attempt, there are still the following disadvantages: (1) the impact of the flight altitude of the UAV on energy consumption wasn't be considered; (2) the impact of the maximum number of IoT devices sending data to the UAV simultaneously on energy consumption wasn't be studied; (3) DEVIPS needs to set crossover rate and mutation rate, which will restrict its applications.

To overcome the aforementioned three disadvantages of DEVIPS, this paper proposes a novel population-based optimization method, namely the backtracking search algorithm with a dynamic population (BSADP), to optimize the path planning of the UAV served as the IoT data collection platform. The main contributions of this paper are as follows:

- This paper researches a UAV-based IoT data collection system from the perspective of energy consumption. The goal of this paper is to find the optimal number and locations of the stop points simultaneously to achieve the minimum energy consumption of this system.
- A new metaheuristic method called BSADP is proposed. The most notable feature of BSADP is that it doesn't rely on any control parameters. Therefore, BSADP has great potential to be applied to solve different optimization problems.
- A population adjustment mechanism based on opposition-based learning is designed. In this mechanism, five candidate adjustment operations including two insert operations, two replace operations and one delete operations are built. The insert operation, replace operation and delete operation mean adding, keeping and reducing the number of stop points, respectively.
- This paper studies the considered system with two energy consumption formulations to verify the optimization performance of BSADP.

The rest of this paper is organized as follows. Section 2 introduces two energy consumption formulations of the considered UAV-based IoT data collection system. Section 3 describes the framework and implementation of BSADP. Experimental results and discussion are presented in Section 4. Lastly, the conclusion and further work are given in Section 5.

2 Problem statement

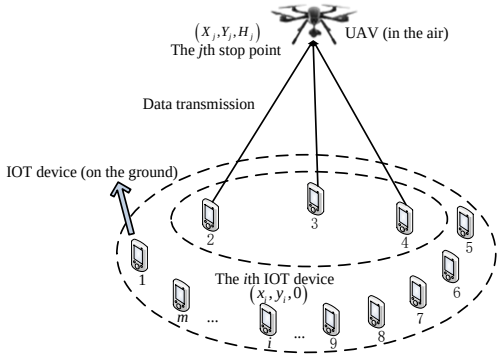


Fig. 1. A UAV-based IoT data collection system.

Fig. 1 presents the considered UAV-based IoT data collection system (Huang et al., 2019), which can be described as follows. From Fig. 1, this system includes a hovering UAV at one stop point and a set of IoT devices on the ground. The UAV is employed as a data collection platform, which is collecting data from the 2nd, 3rd, and 4th IoT devices. The goal of this paper is to minimize the

energy consumption of the system in the process of collecting data. Next, the energy consumption formulation of the system is deduced, which has two types, i.e. energy consumption formulation-I (ECF-I) and energy consumption formulation-II (ECF-II).

2.1 ECF-I

As can be seen from Fig. 1, the number of IoT devices is m and the location of the ith $(i \in [1,m])$ IoT device is denoted as $(x_i, y_i, 0)$, where x_i and y_i are the coordinates of the values in the x-axis and y-axis of the location of the ith IoT device, respectively. Assume the number of stop points is n and the location of the UAV at the jth $(j \in [1,n])$ stop point is denoted as (X_j, Y_j, H_j), where X_j, Y_j and H_j represent the coordinates of the values in the x-axis, y-axis and z-axis of the location of the jth stop point. Euclidean distance d_{ij} between the ith IoT device and the jth stop point of the UAV can be denoted as

$$d_{ij} = \sqrt{(X_j - x_i)^2 + (Y_j - y_i)^2 + (H_j - 0)^2}, \ i \in [1,m], j \in [1,n] \tag{1}$$

As done in (Huang et al., 2019), a binary variable c_{ij} is used to refer to the connection status between the ith IoT device and the UAV at the jth stop point. Moreover, c_{ij} is equal to 0 or 1, which means close or open the connection, respectively. Note that only if a connection is open, an IoT device is allowed to send data to the UAV. To minimize the energy consumed in transmitting data, each candidate IoT device chooses the nearest stop point to send data. c_{ij} can be expressed as

$$c_{ij} = \begin{cases} 1, \text{ if } j = \arg \min\limits_{j \in 1,2,\ldots,n} d_{ij}, i \in [1,m] \\ 0, \text{otherwise} \end{cases} \tag{2}$$

To ensure that all the IoT devices can transmit data to the UAV once, c_{ij} should meet the following constraint

$$\sum_{j=1}^{n} c_{ij} = 1, \ i \in [1,m] \tag{3}$$

In addition, to make full use of system bandwidth, at most β IoT devices can send data at the same time to the UAV at the jth stop point. Thus the following constraint should be met

$$\sum_{i=1}^{m} c_{ij} \le \beta, \ j \in [1,n] \tag{4}$$

To ensure that each IoT device can be connected to the UAV, c_{ij} should meet the following condition

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} = m \tag{5}$$

The channel power gain G_{ij} between the UAV at the jth stop point and the ith IoT device can be expressed as (Zhan et al., 2018; Huang et al., 2019)

$$G_{ij} = G_0 d_{ij}^{-2} = \frac{G_0}{(X_j - x_i)^2 + (Y_j - y_i)^2 + (H_j - 0)^2}, i \in [1,m], j \in [1,n] \tag{6}$$

where G_0 is the channel power gain at the reference distance $d_0 = 1m$, where the line-of-sight condition is assumed. Then, the data rate R_{ij} between the ith IoT device sends data to the UAV at the jth stop point is computed by (Huang et al., 2019)

$$R_{ij} = B \log_2^{1 + \frac{p_i G_{ij}}{\delta^2}}, i \in [1,m], j \in [1,n] \tag{7}$$

where B is the system bandwidth, p_i is the transmitting power between the ith IoT device and the UAV, and δ^2 is the white Gaussian noise power. Thus when the ith IoT device sends L_i amount of data to the UAV at the jth stop point, the corresponding energy consumption of the IoT device can be represented as

$$E_{ij} = \frac{p_i L_i}{R_{ij}}, i \in [1,m], j \in [1,n] \tag{8}$$

Obviously, the total energy consumed by all IoT devices can be expressed as

$$E_{\text{IOT}} = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} E_{ij}, i \in [1,m], j \in [1,n] \tag{9}$$

Assume that the UAV at the jth stop point will be settled into a hover for some time T_j to collect data. The hover time can be written as (Huang et al., 2019)

$$T_j = \max_{i\in[1,m]}\left\{c_{ij}\frac{L_i}{R_{ij}}\right\}, j \in [1,n] \tag{10}$$

Therefore, the energy consumed by the UAV at the jth stop point $E_{\text{uav},j}$ is denoted as

$$E_{\text{uav},j} = p_{\text{uav,h}} T_j, j \in [1,n] \tag{11}$$

where $p_{\text{uav,h}}$ indicates the hover power of the UAV. Then, the total energy consumed by the UAV can be written as

$$E_{\text{UAV}} = \sum_{j=1}^{n} E_{\text{uav},j} \tag{12}$$

Based on the above analysis, the problem of minimizing the energy consumption of the system can be formulated as

$$\begin{aligned} &\min_{\{X_j,Y_j,H_j\},n} \quad E_{\text{UAV}} + \lambda E_{\text{IOT}} \\ &\text{Subject to} \\ &X_{\min} \le X_j \le X_{\max}, j \in [1,n] \\ &Y_{\min} \le Y_j \le Y_{\max}, j \in [1,n] \\ &H_{\min} \le H_j \le H_{\max}, j \in [1,n] \\ &n_{\min} \le n \le n_{\max} \end{aligned} \tag{13}$$

where λ is the weight between the energy consumption of the UAV and that of IoT devices, which meets $\lambda \ge 0$; $X_{\min}$ and $X_{\max}$ are the lower and upper limits of X_j, respectively; $Y_{\min}$ and $Y_{\max}$ are the lower and upper limits of Y_j, respectively; $H_{\min}$ and $H_{\max}$ are the lower and upper limits of H_j, respectively; $n_{\min}$ and $n_{\max}$ are the lower and upper limits of n, respectively. As shown in Eq. (13), minimizing the energy consumption of the system is to find the optimal locations and number of stop points following the given constraints.

2.2 ECF-II

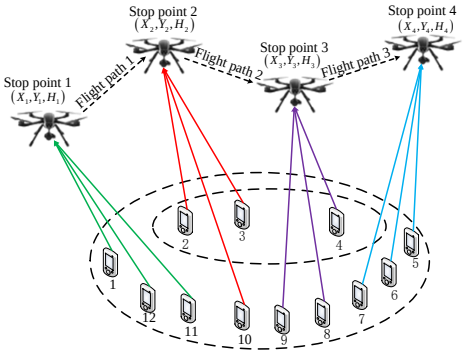


Fig. 2. A simple schematic for the flight path planning of the UAV.

The energy consumption formulation presented in Eq. (13) is the same as (Huang et al., 2019), which doesn't consider the energy consumption produced by the flight of the UAV. To complete the data collection of all the IoT devices, the UAV has multiple stop points in the designed flight path as shown in Fig. 2. As can be seen from Fig. 2, when flying from one stop point to the next, the UAV will consume some energy. Motivated by this, to obtain the energy consumption of the considered system more precisely, this paper defines a new energy consumption formation, which can be written by

$$\begin{aligned} &\min_{\{X_j,Y_j,H_j\},n} \quad E_{\text{UAV}} + \lambda E_{\text{IOT}} + E_{\text{Flight}} \\ &\text{Subject to} \\ &X_{\min} \le X_j \le X_{\max}, j \in [1,n] \\ &Y_{\min} \le Y_j \le Y_{\max}, j \in [1,n] \\ &H_{\min} \le H_j \le H_{\max}, j \in [1,n] \\ &n_{\min} \le n \le n_{\max} \end{aligned} \tag{14}$$

where E_{Flight} is the energy consumption of the flight path of the UAV. For the convenience of computing, the UAV moves with uniform speed in a straight line between two stop points. Thus, E_{Flight} can be computed by

$$E_{\text{Flight}} = \sum_{j=1}^{n-1} p_{\text{uav,f}} \frac{\left| S_{j+1} - S_j \right|}{v_{\text{uav}}} \tag{15}$$

where $p_{\text{uav,f}}$ is the energy consumption power of the UAV during the flight, v_{uav} is the speed of the UAV, S_{j+1} is the position of the UAV at stop point j+1, and S_j is the position of the UAV at stop point j.

The objective functions in Eqs. (13-14) are to find the minimum energy consumption of the considered system by achieving the optimal number and locations of the stop points. Let $\boldsymbol{X}_{\text{Best}}$ denotes the optimal locations of the stop points of the UAV. If the obtained optimal number of the stop points is κ, $\boldsymbol{X}_{\text{Best}}$ can be expressed by $\boldsymbol{X}_{\text{Best}} = \left[(X_1, Y_1, H_1), (X_2, Y_2, H_2), \ldots, (X_\kappa, Y_\kappa, H_\kappa) \right]$. According to the characteristics of the objective functions in Eqs. (13-14) and the optimal locations of the stop points, adopting population-based optimization algorithms are the appropriate choice based on the following consideration. $\boldsymbol{X}_{\text{Best}}$ can be seen as a population of κ individuals. Each element in the $\boldsymbol{X}_{\text{Best}}$ can be regarded as an individual of the population. Inspired by this, BSADP is proposed to solve the optimization problems shown in Eqs. (13-14), which is introduced in Section 3.

3 Proposed method

3.1 Motivation

Population size and terminal condition are two essential parameters for each population-based optimization algorithm. However, most of the reported population-based optimization algorithms need extra control parameters reflecting their characteristics. Take differential evolution (DE) (Houssein et al., 2022; Marinho et al., 2021; Tao et al., 2021) as an example. DE is a classical population-based optimization algorithm and has two extra control parameters, i.e. crossover rate and mutation rate. Note that, although DE has good global searchability, it faces the challenge of how to select the most suitable values of the two extra control parameters in solving different problems with different characteristics. Clearly, this challenge will restrict the applications of DE.

The backtracking search algorithm (BSA) is one of the latest population-based optimization algorithms, which was developed by Civicioglu (Civicioglu, 2013). BSA is a simple and efficient optimization algorithm, which consists of the following components:

(1) Initialization

In BSA, the population $\boldsymbol{X} = \left\{ \boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D}), i = 1, 2, \ldots, N \right\}$ is initialized by

$$x_{i,j} = \boldsymbol{l}_j + \gamma \cdot (\boldsymbol{u}_j - \boldsymbol{l}_j), \quad i = 1, 2, \mathrm{K}, N, \; j = 1, 2, \mathrm{K}, D \tag{16}$$

where N is the population size, D is the number of variables, $\boldsymbol{u}_j$ is the upper limit of the jth variable and $\boldsymbol{l}_j$ is the lower limit of the jth variable. The historical population $\boldsymbol{X}_{\text{old}} = \left\{ \boldsymbol{x}_{\text{old},1}, \boldsymbol{x}_{\text{old},2}, \ldots, \boldsymbol{x}_{\text{old},N} \right\}$ is also initialized by Eq. (18).

(2) Selection-I

The objective of this operator is to regenerate the historical population $\boldsymbol{X}_{\text{old}}$ by

$$\boldsymbol{X}_{\text{old}} = \begin{cases} \boldsymbol{X}, \text{if } \lambda_1 < \lambda_2 \\ \boldsymbol{X}_{\text{old}}, \text{otherwise} \end{cases} \tag{17}$$

where λ_1 and λ_2 are two random numbers uniformly distributed over $[0,1]$. Next, $\boldsymbol{X}_{\text{old}}$ is further processed by

$$\boldsymbol{X}_{\text{old}} = \text{permuting}(\boldsymbol{X}_{\text{old}}) \tag{18}$$

where permuting is a random shuffling function.

(3) Mutation

Mutation is to generate the initial form of the trial population, which can be expressed as

$$\boldsymbol{v}_{\text{initial},i} = \boldsymbol{x}_i + F \cdot (\boldsymbol{x}_{\text{old},i} - \boldsymbol{x}_i) \tag{19}$$

where F is used to control the amplitude of the search-direction matrix. In addition, F is equal to $3 \cdot \mu$ (μ is a rand number obeying standard normal distribution) (Civicioglu, 2013).

(4) Crossover

Crossover operator is to achieve the final form of the trial population $\boldsymbol{V}$ ($\boldsymbol{V}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,\mathrm{K},\boldsymbol{v}_N\}$) by a binary integer-valued matrix $\boldsymbol{M}=\{\boldsymbol{m}_1,\boldsymbol{m}_2,\mathrm{K},\boldsymbol{m}_N\}$, which can be written as

$$\boldsymbol{v}_i=\boldsymbol{x}_i+F\cdot\boldsymbol{m}_i\cdot(\boldsymbol{x}_{\mathrm{old},i}-\boldsymbol{x}_i) \tag{20}$$

In addition, the method of generating the binary integer-valued matrix $\boldsymbol{M}$ is the same as that of BSA (Civicioglu, 2013). Matrix $\boldsymbol{M}$ is adjusted by an extra control parameter mix rate m_{rate}.

(5) Selection-II

This operator is to select the better individuals from the current population $\boldsymbol{X}$ and the obtained trial population $\boldsymbol{V}$, which is denoted as

$$\boldsymbol{x}_i=\begin{cases}\boldsymbol{v}_i, & \text{if } f(\boldsymbol{v}_i)<f(\boldsymbol{x}_i)\\ \boldsymbol{x}_i, & \text{otherwise}\end{cases} \tag{21}$$

According to the components of BSA, BSA has only one extra control parameter, i.e. mix rate. Thus BSA has the potential to be transformed into an algorithm without extra control parameters. With the increasing number of iterations, individuals of the population gradually gather towards the obtained current optimal solution, which may lead to the loss of population diversity. Thus, once the candidate solutions are trapped in the local optimal solutions, it is very difficult to escape. Obviously, how to keep population diversity is an inevitable problem for all population-based optimization algorithms. Note that, the multi-search mechanism has been proved to be an effective way to keep population diversity (Wang et al., 2019; Xia et al., 2019; Zhao et al., 2021). As a new technique, opposition-based learning (OBL) has been widely applied to improve the performance of many population-based algorithms (Fan et al., 2021; Sarkhel et al., 2018; Yang et al., 2022). The core idea of OBL is to find a closer (fitter) solution to the current solution and its opposite solution, which has the potential to accelerate the convergence of population-based optimization algorithms (Rahnamayan et al., 2008). Given this feature of OBL, OBL can be used to build the multi-search mechanism.

3.2 BSADP

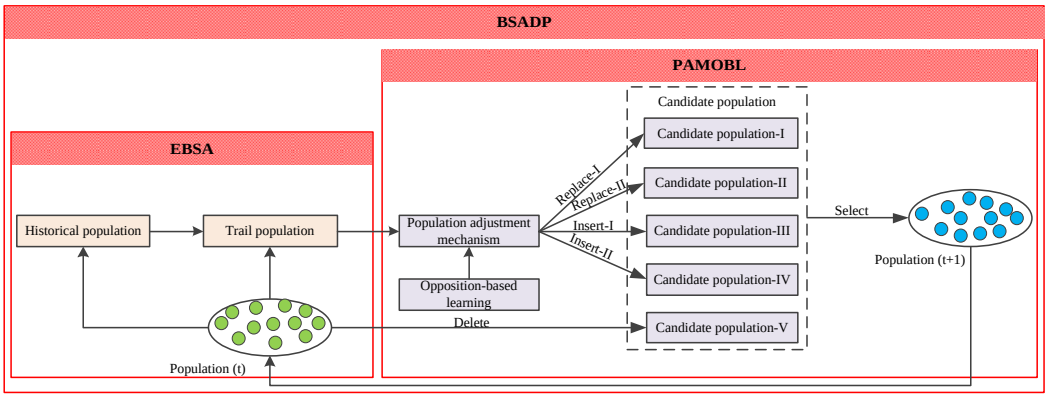


Fig. 3. The framework of the proposed BSADP.

Fig. 3 presents the framework of BSADP. As can be seen from Fig. 3, BSADP has a very simple structure, which includes two core components, i.e. enhanced backtracking search (EBSA) and population adjustment mechanism with opposition-based learning (PAMOBL). From Fig. 3, the implementation of BSADP can be briefly stated as follows. Firstly, EBSA is performed to achieve the trail population. Then, PAMOBL is employed to generate the candidate population. Lastly, the optimal population is selected from the candidate population. The characteristics of BSADP can be stated as follows:

(1) Unlike most of the reported population-based optimization algorithms, BSADP only depends on the essential population size and terminal condition for optimization. In addition, when BSADP is used to optimize the locations and the number of stop points, population size is equal to the number of stop points which means population size is eliminated. Thus, BSADP doesn't consider the issue of parameter settings, which has a great potential to be widely used to solve different types of optimization problems.

(2) BSADP has a very simple framework and doesn't refer to complex computation operations. Clearly, BSADP is an efficient optimization method and doesn't require more computational effort.

(3) BSADP is based on EBSA and PAMOBL. In EBSA and PAMOBL, the strategies for keeping population diversity are introduced. More specifically, EBSA uses the current population and the historical population to produce the next generation population, which can be found in Section 3.2.1. PAMOBL employs five different operators to generate a candidate population, which is described in Section 3.2.2.

3.2.1 The proposed EBSA

As mentioned in Section 3.1, BSA only has an extra control parameter, i.e. mix rate. Although the good global search ability of BSA has been proven (Civicioglu, 2013), BSA doesn't make full use of the population information and only relies on the historical population to generate the next generation population. Thus, when BSA is employed to solve some complex optimization problems, BSA may be trapped in the local optimal solutions. To overcome this disadvantage of BSA, EBSA is proposed. The improved strategies in EBSA can be stated as follows:

(1) Increase the population diversity. To make full use of population information, EBSA employs the historical population and the current population to generate the next generation population. The mutation operator is defined as

$$\boldsymbol{v}_{\text{initial},i} = \boldsymbol{x}_i + F \cdot (\frac{\boldsymbol{x}_{\text{old},i} - \boldsymbol{x}_i + \boldsymbol{x}_k - \boldsymbol{x}_i}{2}) \tag{22}$$

where k is a random integer between 1 and N. Note that, k is not equal to i. To better show the advantage of the designed strategy, Eq. (22) can be rewritten as

$$\boldsymbol{v}_{\text{initial},i} = \boldsymbol{x}_i + 0.5 \cdot F \cdot (\boldsymbol{x}_{\text{old},i} - \boldsymbol{x}_i) + 0.5 \cdot F \cdot (\boldsymbol{x}_k - \boldsymbol{x}_i) \tag{23}$$

In Eq. (23), the third term on the right can be regarded as a random perturbed term, which is conducive to keep population diversity. In addition, the parameter F is a random number subject to the standard normal distribution.

(2) Eliminate the mix rate. Mix rate is used to control the binary integer-valued matrix $\boldsymbol{M}$. Further, crossover probability increases with the mix rate increasing. For different problems with different characteristics, the mix rate should be adjusted. In EBSA, we replace the binary-valued matrix with a random crossover factor $\boldsymbol{C}=[C_1, C_2, \mathrm{K}, C_N]$. Crossover operator in EBSA can be expressed as

$$\boldsymbol{v}_{\text{initial},i} = \boldsymbol{x}_i + F \cdot C_i \cdot (\frac{\boldsymbol{x}_{\text{old},i} - \boldsymbol{x}_i + \boldsymbol{x}_k - \boldsymbol{x}_i}{2}) \tag{24}$$

where C_i is a random number between 0 and 1.

3.2.2 The proposed PAMOBL

Inspired by DEVIPS, we also first assume that the population size is equal to the number of stop points. Then, the optimal number of stop points is found by constantly adjusting the population size. Note that, adjusting population size in the search process also can be regarded as a way of keeping population diversity. To achieve this, PAMOBL is built, which includes five adjustment strategies, i.e. replace operator-I, replace operator-II, insert operator-I, insert operator-II and delete operator. The adjustment strategies in PAMOBL are based on the current population $\boldsymbol{X}$, the trail population $\boldsymbol{v}$ generated by EBSA, and opposition-based learning, which can be stated as follows:

(1) Replace operator-I. This operator first uses the trail individual $\boldsymbol{v}_i (i = 1, 2, \mathrm{K}, N)$ to replace one randomly selected individual $\boldsymbol{x}_s (s = 1, 2, \mathrm{K}, N)$ of $\boldsymbol{X}$ and then the candidate population-I $\boldsymbol{X}_i^{\mathrm{I}}$ is generated.

(2) Replace operator-II. One candidate individual $\boldsymbol{p}$ is generated by opposition-based learning, which can be expressed as

$$\boldsymbol{p} = \boldsymbol{v}_{\max} + \boldsymbol{v}_{\min} - \boldsymbol{v}_i \tag{25}$$

where $\boldsymbol{v}_{\max}$ and $\boldsymbol{v}_{\min}$ are the upper limit and the lower limit of $\boldsymbol{v}$, respectively. The candidate individual $\boldsymbol{p}$ replaces one randomly selected individual $\boldsymbol{x}_s (s = 1, 2, \mathrm{K}, N)$ of $\boldsymbol{X}$ and then the candidate population-II $\boldsymbol{X}_i^{\mathrm{II}}$ is generated.

(3) Insert operator-I. This operator first adds the trail individual $\boldsymbol{v}_i (i = 1, 2, \mathrm{K}, N)$ to $\boldsymbol{X}$ and then the candidate population-III $\boldsymbol{X}_i^{\mathrm{III}}$ is generated.

(4) Insert operator-II. The candidate individual $\boldsymbol{p}$ produced by Eq. (25) is added to $\boldsymbol{X}$ and then the candidate population-IV $\boldsymbol{X}_i^{\mathrm{IV}}$ is generated.

(5) Delete operator. One randomly selected individual $\boldsymbol{x}_s (s = 1,2,\mathrm{K}, N)$ is deleted from $\boldsymbol{X}$ and then the candidate population-V $\boldsymbol{X}_i^{\mathrm{V}}$ is generated.

After performing the adjustment strategies, the best candidate population $\boldsymbol{X}_{\mathrm{opt},i}$ is selected from $\boldsymbol{X}_i^{\mathrm{I}}$, $\boldsymbol{X}_i^{\mathrm{II}}$, $\boldsymbol{X}_i^{\mathrm{III}}$, $\boldsymbol{X}_i^{\mathrm{IV}}$ and $\boldsymbol{X}_i^{\mathrm{V}}$. Then the best population $\boldsymbol{X}_{\mathrm{opt}}$ can be selected from $\boldsymbol{X}_{\mathrm{opt},i}$. The flowchart of PAMOBL has been shown in **Algorithm 1**.

Algorithm 1 The proposed PAMOBL.

Input: Population size N, the current population $\boldsymbol{X}$ and the trail population $\boldsymbol{v}$

01: **For** $i = 1:N$
02: Perform replace operator-I for $\boldsymbol{X}$ and generate the candidate population-I $\boldsymbol{X}_i^{\mathrm{I}}$
03: Perform replace operator-II for $\boldsymbol{X}$ via Eq. (25) and generate the candidate population-II $\boldsymbol{X}_i^{\mathrm{II}}$
04: Perform insert operator-I for $\boldsymbol{X}$ and generate the candidate population-III $\boldsymbol{X}_i^{\mathrm{III}}$
05: Perform insert operator-II for $\boldsymbol{X}$ via Eq. (25) and generate the candidate population-IV $\boldsymbol{X}_i^{\mathrm{IV}}$
06: Perform delete operator for $\boldsymbol{X}$ and generate the candidate population-V $\boldsymbol{X}_i^{\mathrm{V}}$
07: find the optimal candidate population $\boldsymbol{X}_{\mathrm{opt},i}$ from $\boldsymbol{X}_i^{\mathrm{I}}$, $\boldsymbol{X}_i^{\mathrm{II}}$, $\boldsymbol{X}_i^{\mathrm{III}}$, $\boldsymbol{X}_i^{\mathrm{IV}}$ and $\boldsymbol{X}_i^{\mathrm{V}}$.
08: **End for**
09: Select the optimal population $\boldsymbol{X}_{\mathrm{opt}}$ from $\boldsymbol{X}$ and $\boldsymbol{X}_{\mathrm{opt},i}, i = 1,2,\mathrm{K}, N$

Output: The optimal population $\boldsymbol{X}_{\mathrm{opt}}$

Algorithm 2 The proposed BSADP

Input: Population size N, the number of IoT devices m, the upper limits of variables $\boldsymbol{U}$, the lower limits of variables $\boldsymbol{L}$, the current number of iterations $t = 0$ and the maximum number of iterations $t_{\max}$.

/* Initialization */
01: Initialize N by $N = m$
02: Initialize population $\boldsymbol{X}$ and $\boldsymbol{X}_{\mathrm{old}}$ via Eq. (16), Eq. (26), and Eq. (27)
03: Calculate the fitness value of every individual and achieve the optimal solution $\boldsymbol{x}_{\mathrm{opt}}$
/* Main loop */
04: **While** $t < t_{\max}$ **do**
05: Generate the random crossover factor $\boldsymbol{C}$ and adjust F
/* Selection-I */
06: Regenerate the historical population $\boldsymbol{X}_{\mathrm{old}}$ via Eqs. (17-18)
07: **For** $i = 1:N$
/* Mutation */
08: Generate the initial form of the trial individual $\boldsymbol{v}_{\mathrm{initial},i}$ via Eq. (24)
/* Crossover */
09: Generate the final form of the trial individual $\boldsymbol{v}_i$ via Eq. (21)
/* Population adjustment mechanism */
10: Generate candidate population $\boldsymbol{X}_i^{\mathrm{I}}$, $\boldsymbol{X}_i^{\mathrm{II}}$, $\boldsymbol{X}_i^{\mathrm{III}}$, $\boldsymbol{X}_i^{\mathrm{IV}}$ and $\boldsymbol{X}_i^{\mathrm{V}}$ according to **Algorithm 1**
11: Select the optimal candidate population $\boldsymbol{X}_{\mathrm{opt},i}$
12: **End for**
13: Select $\boldsymbol{X}_{\mathrm{opt}}$ from $\boldsymbol{X}$ and $\boldsymbol{X}_{\mathrm{opt},i}, i = 1,2,\ldots,N$
14: $t = t + 1$
15: **End while**

Output: The optimal solution $\boldsymbol{X}_{\mathrm{opt}}$

3.2.3 The implementation of BSADP

As shown in Eqs. (13-14), solving the energy consumption problem of the UAV-assisted IoT data collection system is to find the optimal number and positions of stop points of the UAV. Clearly, the number and positions of the stop points of the UAV are the variables of the considered problem, which can be denoted by

$$\boldsymbol{L} = [X_{\min}\ Y_{\min}\ H_{\min}] \tag{26}$$

$$\boldsymbol{U} = [X_{\max}\ Y_{\max}\ H_{\max}] \tag{27}$$

$$1 \le n \le m \tag{28}$$

where $\boldsymbol{L}$ is the lower limit of the positions of the stop points, $\boldsymbol{U}$ is the upper limit of the positions of the stop points, and the number of stop points n is not more than the number of IoT devices m. Note that, n is the population size and is initialized by m.

Algorithm 2 presents the flowchart of BSADP. As can be seen from **Algorithm** 2, BSADP has a simple structure and is easy to work. In addition, BSADP is an algorithm without any effort for fine-tuning initial parameters, which has great potential to be used for solving the energy consumption problems of a UAV-assisted IoT data collection system with different scenarios. From **Algorithm 2**, the computational complexity of BSADP mainly includes position update and comparison of fitness values, which can be described as follows.

- Position update. In one loop, each individual with D elements generates its trail individual, which can be found in line 8 of **Algorithm 2**. In addition, replace operator-II and insert operator-II also generate two individuals by Eq. (25). Thus, the complexity of the position update can be written as $O(3NDt_{\max})$, where $t_{\max}$ is the maximum number of iterations.
- Comparison of fitness value. In BSADP, the comparison of fitness values is performed six times. The first time is in line 9 of **Algorithm 2** and the rest five times are in line 10 of **Algorithm 2**. Thus, the complexity of the comparison of fitness values is $O(6Nt_{\max})$.

Based on the above discussion, the computational complexity of BSADP is $O(6Nt_{\max}+3NDt_{\max})$. Note that, the population size in BSADP is dynamic. Therefore, the computational complexity of BSADP can be further represented by

$$\mathrm{CC}_{\mathrm{BSADP}} = O(\sum_{i=1}^{t_{\max}} 6N_i t_{\max} + 3N_i D t_{\max}) \tag{29}$$

where $\mathrm{CC}_{\mathrm{BSADP}}$ denotes the computational complexity of BSADP and N_i means the population size in the ith iteration.

4 Experimental studies

4.1 Experimental preparation

According to the authors of DEVIPS, they are the first attempt to optimize the locations and the number of the stop points simultaneously to minimize the energy consumption of a UAV-based IoT data collection system. In addition, the variable population size strategy in DEVIPS motivates us to design PAMOBL. Thus, the designed experiments emphatically focus on the comparisons with DEVIPS. Besides, we also compare BSADP with some recently reported population-based algorithms that preset the number of the stop points. The parameters of the compared algorithms were extracted from the original references, which are described as follows:

(1) JADE (Zhang & Sanderson, 2009). JADE is a very popular variant of DE. For JADE, the population size was set to 100. In solving the deployment of UAV, JADE needs to preset the number of the stop points.

(2) DEEM (Wang et al., 2018). DEEM is a recently proposed population-based optimization algorithm. Like BSADP and DEVIPS, DEEM uses the population to represent the entire deployment of the UAV. For DEEM, population size, crossover probability and mutation probability were set to 100, 0.9 and 0.9, respectively. Like JADE, DEEM also needs to preset the number of stop points in solving the deployment of the UAV.

(3) DEVIPS (Huang et al., 2019). For DEVIPS, crossover probability and mutation probability were set to 0.6 and 0.5, respectively. In addition, the population size was equal to the number of IoT devices.

(4) BSADP. BSADP doesn't have any other extra control parameters. In addition, population size was equal to the number of IoT devices.

In addition, the maximum number of function evaluations was set to 100,000 as done in (Huang et al., 2019). Each algorithm was independently run 30 times for each test case. The parameters of UAV and IoT devices in this paper have been listed in Table 1. Note that, for the parameters in Table 1, $p_{\mathrm{uav,f}}$ and v_{uav} are set to address ECF-II and the other parameters were the same as those in (Huang et al., 2019). To verify the performance of BSADP for the energy consumption of the UAV-based IoT data collection system, a series of experiments are executed. Table 2, Table 3, Table 4, Table 5, and Table 6 present the experimental results. In the five tables, "Mean" and "Std" denote mean value and standard deviation, respectively. The best results were highlighted in bold in the five tables. To better show the performance of BSADP, the average results achieved of 30 runs from each algorithm

are subjected to Wilcoxon signed rank test (Derrac et al., 2011) with a significant level α=0.05. Experimental results of Wilcoxon signed rank test are shown in Tables 3-6. In the two tables, "+" means that BSADP outperforms the compared algorithm; "=" indicates that there is no significant difference between BSADP and the compared algorithm; "-" demonstrates that BSADP is inferior to the compared algorithm. The last columns of the four tables offer the total test results. Besides, the locations of IoT devices and the amount of transmission data from IoT devices to the UAV can be loaded from the following website: https://intleo.csu.edu.cn/publication.html.

Table 1
Parameter settings of the designed system.

Parameter	Value
Area for random IoT devices distribution (S×S)	1000[m]×1000[m]
The mount of transmission data $L_i(i=1,2,\mathrm{K},m)$	[1,1000][MB]
The transmitting power $p_i(i=1,2,\mathrm{K},m)$	0.1[W]
The channel power gain h_0	-25[dB]
The white Gaussian noise power δ^2	-250[dB]
The system bandwidth B	1[MHz]
The hover power of the UAV $p_{\mathrm{uav,h}}$	1000[W]
The flight power of the UAV $p_{\mathrm{uav,f}}$	1000[W]
The flight speed of the UAV v_{uav}	40[Km/h]
The weight value λ	1000
The total number of IoT devices m	[100,700]

Table 2
Experimental results obtained by JADE, DEEM, DEVIPS and BSADP.

m	JADE		DEEM		DEVIPS		BSADP	
	Mean (J)	Std Dev (J)	Mean (J)	Std Dev (J)	Mean (J)	Std Dev (J)	Mean (J)	Std Dev (J)
100	1.4837E+6	6.5274E+3	1.3507E+6	1.6116E+4	1.2525E+6	6.7254E+3	**1.2492E+6**	6.1467E+3
200	2.9912E+6	1.4947E+4	2.7035E+6	2.2978E+4	2.5045E+6	7.1329E+3	**2.5003E+6**	9.8533E+3
300	4.3166E+6	1.3839E+4	3.8755E+6	2.7542E+4	3.5809E+6	1.4834E+4	**3.5788E+6**	1.1307E+4
400	6.0787E+6	1.6834E+4	5.3782E+6	3.0406E+4	5.0016E+6	1.8278E+4	**4.9937E+6**	2.0658E+4
500	7.4686E+6	1.4416E+4	6.5991E+6	3.4363E+4	6.1248E+6	1.8445E+4	**6.1143E+6**	1.8183E+4
600	9.2240E+6	1.9429E+4	8.0769E+6	5.5218E+4	7.5628E+6	2.0203E+4	**7.5408E+6**	2.1798E+4
700	1.0434E+7	2.5772E+4	9.1732E+6	4.0661E+4	8.5702E+6	3.5668E+4	**8.5525E+6**	3.1182E+4

4.2 Comparisons of DEEM, JADE, DEVIPS, and BSADP

This section repeats the experiment in (Huang et al., 2019) to compare the performance among BSADP, DEVIPS, JADE and DEEM. In this section, the flight altitude of the UAV and the maximum number of IoT devices sending data to the UAV are set to 200m and 5, respectively. For JADE and DEEM, the preset number of stop points was equal to the number of the IoT devices. The candidate solutions are evaluated by ECF-I and the experiment results of DEVIPS, JADE and DEEM are extracted from (Huang et al., 2019).

Table 2 presents the experimental results obtained by DEEM, JADE, DEVIPS and BSADP. In this experiment, JADE and DEEM need to preset the number of stop points while the number of stop points for BSADP and DEVIPS is dynamic. By looking carefully at Table 2, BSADP and DEVIPS are significantly superior to JADE and DEEM for all test cases in terms of mean energy consumption. This indicates that DEVIPS and BSADP with the dynamic population mechanism are more effective than JADE and DEEM with presetting the number of stop points for solving the energy consumption of the UAV-based IoT data collection system. In addition, looking at Table 2, BSADP outperforms JADP, DEEM and DEVIPS in all test cases in terms of the mean energy consumption, which supports the excellent global search ability of BSADP.

4.3 Comparisons of DEVIPS and BSADP under ECF-I

This section is to compare the performance of DEVIPS and BSADP under ECF-I. To better compare the performance of BSADP and DEVIPS, the energy consumption difference (ECD) is defined, which can be expressed by

$$\mathrm{ECD}=E_{\mathrm{DEVIPS}}-E_{\mathrm{BSADP}} \tag{30}$$

where E_{DEVIPS} and E_{BSADP} are the mean energy consumption obtained by DEVIPS and BSADP under a

given condition, respectively. From Eq. (30), when ECD is more than 0, DEVIPS has worse performance than BSADP; when ECD is less than 0, DEVIPS has a better performance than BSADP; when ECD is equal to 0, DEVIPS has the same performance with BSADP.

4.3.1 The impact of flight altitude of the UAV under ECF-I

This section is to check the impact of the flight altitude of the UAV on the performance of DEVIPS and BSADP under ECF-I. The maximum number of IoT devices sending data to the UAV is set to 10 and the flight altitude of the UAV is set to between 100m and 500m in this experiment.

Experimental results obtained by BSADP and DEVIPS are presented in Table 3. From Table 3, BSADP outperforms DEVIPS for nearly all test cases in terms of mean energy consumption. Based on the test results produced by Wilcoxon signed rank test, BSADP is superior to DEVIPS in sixty percent of test cases. BSADP and DEVIPS have the same performance on the rest of the test cases. By observing Table 3, with the flight altitude of the UAV increment, the energy consumption from BSADP and DEVIPS gradually increases, which can be explained as follows: according to the designed scenario, when the data transfer volume between IoT devices and the UAV is identified, the transmission time increases with the flight altitude of the UAV increment. Thus the consumed hover energy of the UAV and the consumed emission energy of IoT devices will increase, which results in an increment in the total energy consumption.

Fig. 4 shows the obtained ECD according to the data from Table 3. As can be seen from Fig. 4, ECD is more than 0 in all the considered cases, which means BSADP outperforms DEVIPS in these cases without the constraints of the flight altitude.

Table 3
Experimental results obtained by DEVIPS and BSADP under different flight altitudes of UAV (ECF-I).

H (m)	Algorithm	Indicator	m						
			100	200	300	400	500	600	700
100	DEVIPS	Mean (J)	1.0813E+6	2.1564E+6	3.0765E+6	4.2927E+6	5.2610E+6	6.4796E+6	7.3139E+6
		Std (J)	4.7777E+3	6.4361E+3	8.2385E+3	9.9348E+3	1.1356E+4	1.4655E+4	1.5249E+4
	BSADP	Mean (J)	**1.0804E+6**$^{=}$	**2.1554E+6**$^{=}$	**3.0721E+6**$^{+}$	**4.2855E+6**$^{+}$	**5.2483E+6**$^{+}$	**6.4791E+6**$^{=}$	**7.3071E+6**$^{=}$
		Std (J)	4.4739E+3	6.6538E+3	6.8043E+3	9.6568E+3	1.1154E+4	1.3325E+4	1.8488E+4
200	DEVIPS	Mean (J)	1.1207E+6	2.2326E+6	3.1849E+6	4.4444E+6	5.4485E+6	6.7097E+6	7.5872E+6
		Std (J)	4.6189E+3	5.0574E+3	9.1457E+3	1.0805E+4	1.1581E+4	1.4014E+4	2.1129E+4
	BSADP	Mean (J)	**1.1171E+6**$^{+}$	**2.2307E+6**$^{=}$	**3.1792E+6**$^{+}$	**4.4424E+6**$^{=}$	**5.4406E+6**$^{+}$	**6.7021E+6**$^{+}$	**7.5711E+6**$^{+}$
		Std (J)	5.9008E+3	8.4638E+3	7.6264E+3	1.1789E+4	1.3087E+4	1.2130E+4	1.8887E+4
300	DEVIPS	Mean (J)	1.1433E+6	2.2801E+6	3.2543E+6	4.5433E+6	5.5636E+6	6.8577E+6	7.7475E+6
		Std (J)	5.0923E+3	5.0744E+3	9.2218E+3	1.1503E+4	1.1527E+4	1.4755E+4	1.5430E+4
	BSADP	Mean (J)	**1.1413E+6**$^{=}$	**2.2772E+6**$^{+}$	**3.2516E+6**$^{=}$	**4.5360E+6**$^{=}$	**5.5618E+6**$^{=}$	**6.8467E+6**$^{+}$	**7.7389E+6**$^{+}$
		Std (J)	4.4169E+3	5.3449E+3	8.9455E+3	1.2619E+4	1.5236E+4	1.3225E+4	1.6286E+4
400	DEVIPS	Mean (J)	1.1602E+6	2.3173E+6	3.3048E+6	4.6139E+6	5.6564E+6	6.9590E+6	7.8673E+6
		Std (J)	4.8157E+3	7.8190E+3	9.4054E+3	1.3371E+4	1.3091E+4	1.7307E+4	1.5092E+4
	BSADP	Mean (J)	**1.1572E+6**$^{+}$	**2.3125E+6**$^{+}$	**3.2994E+6**$^{+}$	**4.6025E+6**$^{+}$	**5.6465E+6**$^{+}$	**6.9526E+6**$^{=}$	**7.8635E+6**$^{=}$
		Std (J)	4.8933E+3	6.2974E+3	8.8069E+3	1.3382E+4	1.2209E+4	1.6331E+4	1.4385E+4
500	DEVIPS	Mean (J)	1.1752E+6	2.3436E+6	3.3463E+6	4.6688E+6	5.7224E+6	7.0491E+6	7.9689E+6
		Std (J)	5.4773E+3	6.2385E+3	8.3092E+3	1.0044E+4	1.5185E+4	1.2986E+4	1.7120E+4
	BSADP	Mean (J)	**1.1719E+6**$^{+}$	**2.3421E+6**$^{=}$	**3.3385E+6**$^{+}$	**4.6626E+6**$^{+}$	**5.7118E+6**$^{+}$	**7.0371E+6**$^{+}$	**7.9593E+6**$^{=}$
		Std (J)	5.0225E+3	5.8161E+3	7.8075E+3	1.1708E+4	1.6282E+4	1.6883E+4	1.8948E+4
+/=/-			21/14/0						

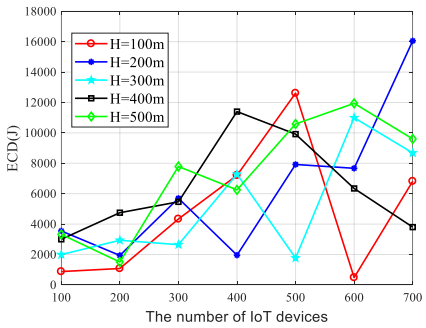


Fig. 4. The obtained ECD according to the data from Table 3.

Table 4
Experimental results obtained by DEVIPS and BSADP under the different numbers of IoT devices sending data to the UAV simultaneously (ECF-I).

β	Algorithm	Indicator	m						
			100	200	300	400	500	600	700
10	DEVIPS	Mean (J)	1.1415E+6	2.2812E+6	3.2511E+6	4.5441E+6	5.5651E+6	6.8496E+6	7.7487E+6
		Std (J)	3.6598E+3	7.9104E+3	8.7189E+3	1.0146E+4	1.5669E+4	1.3815E+4	1.2693E+4
	BSADP	Mean (J)	**1.1410E+6**$^{=}$	**2.2776E+6**$^{=}$	**3.2486E+6**$^{=}$	**4.5357E+6**$^{+}$	**5.5569E+6**$^{=}$	**6.8486E+6**$^{=}$	**7.7414E+6**$^{=}$
		Std (J)	5.2146E+3	7.5440E+3	7.1329E+3	1.1276E+4	1.2242E+4	1.7894E+4	1.7953E+4
20	DEVIPS	Mean (J)	1.0644E+6	2.1194E+6	3.0142E+6	4.2064E+6	5.1497E+6	6.3380E+6	7.1456E+6$^{=}$
		Std (J)	5.1365E+3	5.1653E+3	5.7634E+3	6.6799E+3	9.1444E+3	1.0303E+4	1.0334E+4
	BSADP	Mean (J)	**1.0614E+6**$^{+}$	**2.1174E+6**$^{=}$	**3.0126E+6**$^{=}$	**4.2012E+6**$^{+}$	**5.1468E+6**$^{=}$	**6.3325E+6**$^{=}$	**7.1425E+6**$^{=}$
		Std (J)	4.8782E+3	4.4009E+3	7.0356E+3	6.1640E+3	7.7149E+3	8.3185E+3	9.7089E+3
30	DEVIPS	Mean (J)	1.0414E+6	2.0598E+6	**2.9266E+6**$^{=}$	4.0850E+6	4.9970E+6	6.1477E+6	6.9254E+6
		Std (J)	4.8359E+3	6.0236E+3	6.4543E+3	5.5480E+3	6.6869E+3	7.0691E+3	1.0981E+4
	BSADP	Mean (J)	**1.0346E+6**$^{+}$	**2.0569E+6**$^{=}$	2.9273E+6	**4.0837E+6**$^{=}$	**4.9947E+6**$^{=}$	**6.1426E+6**$^{+}$	**6.9252E+6**$^{=}$
		Std (J)	7.7985E+2	5.8353E+3	6.5956E+3	4.8081E+3	8.2493E+3	8.0648E+3	9.4682E+3
40	DEVIPS	Mean (J)	1.0287E+6	2.0329E+6	2.8851E+6	4.0209E+6	4.9180E+6	6.0506E+6	6.8103E+6
		Std (J)	7.4967E+3	3.3053E+3	7.4912E+3	6.1802E+3	5.8579E+3	7.2865E+3	9.4124E+3
	BSADP	Mean (J)	**1.0188E+6**$^{+}$	**2.0304E+6**$^{+}$	**2.8812E+6**$^{+}$	**4.0185E+6**$^{=}$	**4.9167E+6**$^{=}$	**6.0499E+6**$^{+}$	**6.8100E+6**$^{=}$
		Std (J)	4.4419E+2	2.3959E+3	5.9893E+3	5.9984E+3	7.2583E+3	9.1851E+3	8.6514E+3
50	DEVIPS	Mean (J)	1.0236E+6	2.0174E+6	2.8601E+6	3.9834E+6	4.8736E+6	**5.9903E+6**$^{=}$	6.7407E+6
		Std (J)	7.3982E+3	4.0780E+3	4.5744E+3	5.2657E+3	7.5013E+3	7.8033E+3	8.3110E+3
	BSADP	Mean (J)	**1.0049E+6**$^{+}$	**2.0120E+6**$^{+}$	**2.8560E+6**$^{+}$	**3.9817E+6**$^{+}$	**4.8712E+6**$^{=}$	5.9911E+6	**6.7400E+6**$^{=}$
		Std (J)	5.1258E+3	4.7509E+3	3.4045E+3	5.5311E+3	7.1856E+3	7.0104E+3	7.4525E+3
+/=/-			13/22/0						

4.3.2 The impact of the number of IoT devices sending data to the UAV under ECF-I

This experiment is to investigate the performance of BSADP and DEVIPS with the varying maximum number of IoT devices sending data to UAV under ECF-I. In this experiment, the flight altitude of the UAV is set to 300m and the maximum number of IoT devices sending data to the UAV is set to between 10 and 50.

Table 4 presents the experimental results obtained by DEVIPS and BSADP. As can be seen from Table 4, BSADP can achieve less mean energy consumption than DEVIPS for nearly all test cases. By observing the test results produced by Wilcoxon signed rank test, BSADP is superior to DEVIPS in nearly half of the test cases. BSADP and DEVIPS have the same performance on the rest of the test cases. In addition, when the flight altitude of the UAV and the data transmission volume are fixed, the consumed energy obtained by BSADP and DEVIPS gradually decreases with the number of IoT devices sending data to the UAV simultaneously increasing. This can be explained below: under other fixed conditions, increasing the number of IoT devices sending data to the UAV simultaneously can utilize the communication link between the UAV and the IoT devices more efficiently. However, the excessive number of IoT devices sending data to the UAV simultaneously may lead to network congestion, which will decrease data transmission efficiency.

Fig. 5 presents the obtained ECD according to the data from Table 4. From Fig. 5, the following three conclusions can be made: (1) ECD is less than 0 in two test cases, i.e. $\beta=30, m=300$; $\beta=50, m=600$. That is, DEVIPS is superior to BSADP in the two test cases. (2) ECD is slightly larger than 0 in the three cases, i.e. $\beta=30, m=700$; $\beta=40, m=700$; $\beta=50, m=700$. That is, BSADP has slight advantages over DEVIPS in the three test cases. (3) ECD is significantly larger than 0 in the rest 30 test cases. That is, BSADP has obvious advantages over DEVIPS in the 30 test cases.

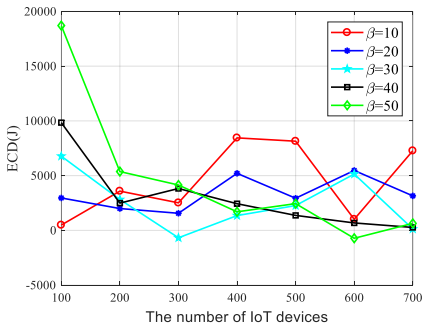


Fig. 5. The obtained ECD according to the data from Table 4.

4.4 Comparisons of DEVIPS and BSADP under ECF-II

4.4.1 The impact of flight altitude of the UAV under ECF-II

This section is to study the impact of flight altitude of UAV on the performance of DEVIPS and BSADP under ECF-II. In this experiment, the maximum number of IoT devices sending data to the UAV and the flight altitude of the UAV are the same as those of Section 4.3.1.

Table 5 shows the experimental results obtained by DEVIPS and BSADP. Looking at Table 5, BSADP can beat DEVIPS in all test cases in terms of mean energy consumption. Based on the test results produced by Wilcoxon signed rank test, DEVIPS only has the same performance as BSADP on one test case ($H=100$, $m=100$) while DEVIPS can't compete with BSADP on the rest 34 test cases. In addition, Fig. 6 displays the obtained ECD according to the data from Table 5. From Fig. 6, ECD is more than 0 in all test cases, which demonstrates BSADP is superior to DEVIPS in all test cases. Besides, with the number of IoT devices increasing, the advantages of BSADP have been on the rise in general, which can be explained as follows. Unlike ECF-I, ECF-II considers the flight path planning of the UAV. There is an increase in the number of IoT devices, which means the UAV needs more stop points to collect the data of all IoT devices. Therefore, the energy consumed by the UAV in the flight path planning will increase. As shown in Eq. (14), an algorithm can find a shorter total flight path of the UAV, which will consume less energy. Clearly, BSADP shows an obvious advantage over DEVIPS in designing the optimal flight path planning of the UAV.

Table 5
Experimental results obtained by DEVIPS and BSADP under different flight altitudes of UAV (ECF-II).

H (m)	Algorithm	Indicator	*m* 100	200	300	400	500	600	700
100	DEVIPS	Mean (J)	1.3258E+6	2.7845E+6	4.2161E+6	5.9389E+6	7.3637E+6	9.1229E+6	1.0445E+7
		Std (J)	5.3890E+4	8.2169E+4	1.1621E+5	1.4858E+5	1.4299E+5	1.4434E+5	2.0050E+5
	BSADP	Mean (J)	**1.3034E+6**$^{=}$	**2.7333E+6**$^{+}$	**4.1410E+6**$^{+}$	**5.8239E+6**$^{+}$	**7.2099E+6**$^{+}$	**8.9185E+6**$^{+}$	**1.0201E+7**$^{+}$
		Std (J)	5.9317E+4	9.4012E+4	8.7058E+4	1.6954E+5	1.3210E+5	1.2081E+5	2.1025E+5
200	DEVIPS	Mean (J)	1.3661E+6	2.8525E+6	4.2958E+6	6.1308E+6	7.5550E+6	9.3068E+6	1.0725E+7
		Std (J)	4.0465E+4	8.0846E+4	1.1011E+5	1.1383E+5	1.3916E+5	1.9612E+5	1.5476E+5
	BSADP	Mean (J)	**1.3189E+6**$^{+}$	**2.8079E+6**$^{+}$	**4.1993E+6**$^{+}$	**5.9272E+6**$^{+}$	**7.3132E+6**$^{+}$	**9.1323E+6**$^{+}$	**1.0456E+7**$^{+}$
		Std (J)	4.7004E+4	7.5315E+4	9.1406E+4	1.1330E+5	1.5725E+5	1.9589E+5	2.0443E+5
300	DEVIPS	Mean (J)	1.3710E+6	2.8999E+6	4.3838E+6	6.2390E+6	7.6803E+6	9.4872E+6	1.0941E+7
		Std (J)	4.2205E+4	7.9033E+4	1.4301E+5	1.2350E+5	1.2060E+5	1.5479E+5	1.9431E+5
	BSADP	Mean (J)	**1.3370E+6**$^{+}$	**2.8445E+6**$^{+}$	**4.2722E+6**$^{+}$	**6.0713E+6**$^{+}$	**7.5552E+6**$^{+}$	**9.2293E+6**$^{+}$	**1.0731E+7**$^{+}$
		Std (J)	4.7242E+4	8.0558E+4	1.3946E+5	1.2119E+5	1.2759E+5	1.5586E+5	1.7436E+5
400	DEVIPS	Mean (J)	1.4148E+6	2.9264E+6	4.4575E+6	6.2541E+6	7.7965E+6	9.5972E+6	1.1061E+7
		Std (J)	6.3135E+4	8.0404E+4	1.0473E+5	1.6002E+5	1.2455E+5	1.3761E+5	1.4920E+5
	BSADP	Mean (J)	**1.3598E+6**$^{+}$	**2.8558E+6**$^{+}$	**4.3639E+6**$^{+}$	**6.1202E+6**$^{+}$	**7.5760E+6**$^{+}$	**9.3628E+6**$^{+}$	**1.0769E+7**$^{+}$
		Std (J)	5.2474E+4	6.7025E+4	1.0578E+5	1.2200E+5	1.3901E+5	1.4144E+5	2.0170E+5
500	DEVIPS	Mean (J)	1.4214E+6	2.9754E+6	4.4945E+6	6.3119E+6	7.8181E+6	9.6572E+6	1.1100E+7
		Std (J)	5.1061E+4	7.1926E+4	8.5158E+4	1.5122E+5	1.3071E+5	2.1110E+5	1.5751E+5
	BSADP	Mean (J)	**1.3763E+6**$^{+}$	**2.9234E+6**$^{+}$	**4.3640E+6**$^{+}$	**6.1896E+6**$^{+}$	**7.6026E+6**$^{+}$	**9.4184E+6**$^{+}$	**1.0845E+7**$^{+}$
		Std (J)	3.7675E+4	7.0190E+4	1.0579E+5	1.3547E+5	1.3871E+5	1.6650E+5	1.4039E+5
+/=/-			34/1/0						

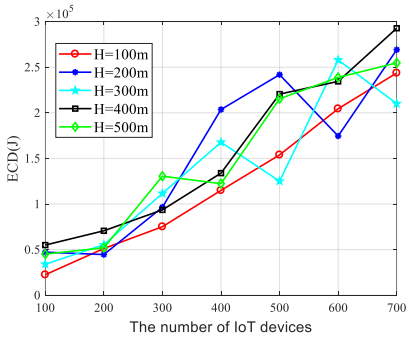


Fig. 6. The obtained ECD according to the data from Table 5.

4.4.2 The impact of the number of IoT devices sending data to the UAV under ECF-II

This experiment is to compare the performance of BSADP and DEVIPS with the varying maximum number of IoT devices sending data to the UAV under ECF-II. In this experiment, the flight altitude of the UAV and the maximum number of IoT devices sending data to the UAV are the same as those of Section 4.3.2.

The experimental results obtained by DEVIPS and BSADP have been presented in Table 6. From Table 6, BSADP can get less mean energy consumption than DEVIPS in all test cases. According to the test results produced by Wilcoxon signed rank test, DEVIPS is inferior to DEVIPS on 32 test cases while DEVIPS only can get the same performance as BSADP in three test cases, i.e. $\beta=30$, $m=100$; $\beta=10$, $m=200$; $\beta=40$, $m=700$. In addition, Fig. 7 shows the obtained ECD according to the data from Table 6. As can be seen from Fig. 7, the following two conclusions can be made: (1) ECD is more than 0 in all test cases, which supports BSADP outperforms DEVIPS in the all test cases; (2) when β is less than 20, the advantages of BSADP are more obvious, which can be explained as follows. β is set to a small value, which means the UAV needs more stop points to complete the data collection task. Thus, more energy needs to be consumed by the UAV in the process of the flight. As mentioned in Section 4.4.1, BSADP has an obvious advantage over DEVIPS in terms of designing the optimal flight path planning, which is also supported by the experimental results of this section.

Table 6
Experimental results obtained by DEVIPS and BSADP under the different number of IoT devices sending data to the UAV simultaneously (ECF-II).

β	Algorithm	Indicator	m						
			100	200	300	400	500	600	700
10	DEVIPS	Mean (J)	1.3996E+6	2.9123E+6	4.4245E+6	6.2378E+6	7.7069E+6	9.4994E+6	1.0945E+7
		Std (J)	4.5859E+4	8.9259E+4	1.0128E+5	1.4459E+5	1.6852E+5	1.6584E+5	1.5977E+5
	BSADP	Mean (J)	**1.3487E+6**$^+$	**2.8718E+6**$^=$	**4.2477E+6**$^+$	**6.0612E+6**$^+$	**7.5037E+6**$^+$	**9.3196E+6**$^+$	**1.0651E+7**$^+$
		Std (J)	4.0994E+4	7.9379E+4	8.8410E+4	1.4747E+5	1.7564E+5	1.9322E+5	1.6742E+5
20	DEVIPS	Mean (J)	1.1436E+6	2.3579E+6	3.4764E+6	4.9267E+6	6.0558E+6	7.4669E+6	8.5733E+6
		Std (J)	2.2637E+4	4.7071E+4	9.1910E+4	1.1166E+5	7.6197E+4	9.4052E+4	1.5126E+5
	BSADP	Mean (J)	**1.1190E+6**$^+$	**2.3096E+6**$^+$	**3.4233E+6**$^+$	**4.8488E+6**$^+$	**5.9820E+6**$^+$	**7.3812E+6**$^+$	**8.4586E+6**$^+$
		Std (J)	1.9973E+4	4.1898E+4	6.5819E+4	8.4323E+4	1.0484E+5	1.1067E+5	1.2649E+5
30	DEVIPS	Mean (J)	1.0221E+6	2.0682E+6	2.9461E+6	4.0810E+6	5.0336E+6	6.2102E+6	7.0186E+6
		Std (J)	1.1469E+4	2.0831E+4	2.5029E+4	3.6529E+4	5.9644E+4	4.8209E+4	5.6632E+4
	BSADP	Mean (J)	**1.0189E+6**$^=$	**2.0440E+6**$^+$	**2.9024E+6**$^+$	**4.0473E+6**$^+$	**4.9849E+6**$^+$	**6.1712E+6**$^+$	**6.9601E+6**$^+$
		Std (J)	1.2137E+4	1.5226E+4	1.7050E+4	2.7497E+4	3.2961E+4	4.2200E+4	5.7413E+4
40	DEVIPS	Mean (J)	1.0631E+6	2.1299E+6	3.0606E+6	4.2846E+6	5.2902E+6	6.5139E+6	7.3769E+6
		Std (J)	1.9723E+4	2.9369E+4	4.3583E+4	4.2913E+4	6.9367E+4	6.4729E+4	7.9366E+4
	BSADP	Mean (J)	**1.0354E+6**$^+$	**2.0847E+6**$^+$	**3.0289E+6**$^+$	**4.2243E+6**$^+$	**5.2366E+6**$^+$	**6.4370E+6**$^+$	**7.3345E+6**$^=$
		Std (J)	8.7707E+3	1.3594E+4	3.7313E+4	4.1179E+4	6.4012E+4	6.3865E+4	7.5843E+4
50	DEVIPS	Mean (J)	1.0317E+6	2.0867E+6	2.9856E+6	4.1395E+6	5.1098E+6	6.3357E+6	7.1767E+6
		Std (J)	1.7718E+4	2.6890E+4	3.2891E+4	3.4992E+4	4.7492E+4	6.2434E+4	6.8308E+4
	BSADP	Mean (J)	**1.0188E+6**$^+$	**2.0579E+6**$^+$	**2.9449E+6**$^+$	**4.1110E+6**$^+$	**5.0879E+6**$^+$	**6.2809E+6**$^+$	**7.0966E+6**$^+$
		Std (J)	1.2240E+4	1.2897E+4	1.8716E+4	3.4814E+4	5.1311E+4	5.1904E+4	5.7642E+4
+/=/-			32/3/0						

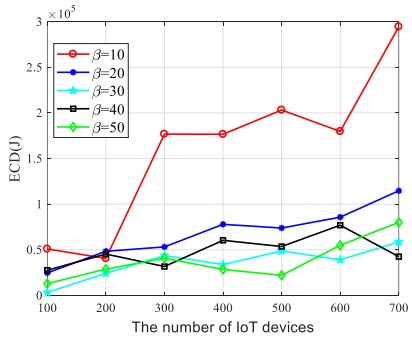
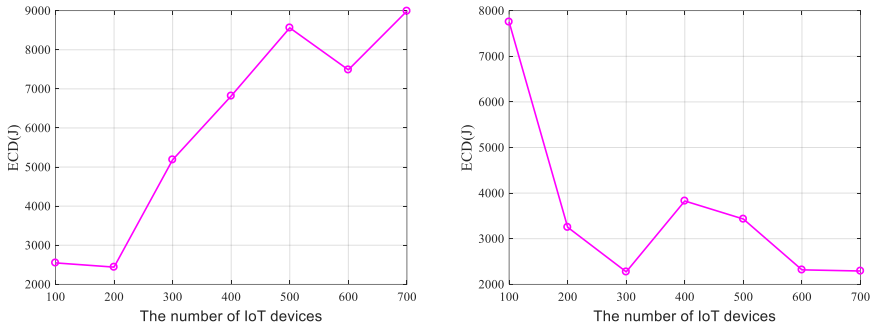


Fig. 7. The obtained ECD according to the data from Table 6.

4.5 Discussion on the effectiveness of the improved strategies in BSADP

This section is to discuss the effectiveness of the improved strategies in BSADP based on the experimental results in Section 4.2, Section 4.3, and Section 4.4.

As mentioned in Section 3.2, the basic idea of BSADP is to keep the population diversity, which is mainly reflected in the following two aspects. On one hand, in the basic BSA, all elements of an individual share the same crossover rate. However, each element of an individual has its own crossover rate in the designed EBSA as shown in Eq. (24). In addition, an individual of the current population is selected to be a direction individual in the designed EBSA as presented in Eq. (23). That is, EBSA has two direction individuals while BSA only has one direction individual. On the other hand, as can be seen from Fig. 3, there are five candidate operators, i.e. two insert operators, one delete operator, and two replace operators, in the built PAMOBL, which are very helpful for adjusting dynamically the population size.



(a) The obtained ECD based on Fig. 4 (b) The obtained ECD based on Fig. 5

(c) The obtained ECD based on Fig. 6 (d) The obtained ECD based on Fig. 7

Fig. 8. The obtained overall ECD according to the data from Figs. 4-7.

According to the experimental results in Section 4.2, in terms of mean energy consumption, DEVIPS and BSADP with variable population sizes are significantly superior to DEEM and JADE with presetting the number of stop points.

Section 4.3 and Section 4.4 focus on comparing the performance between DEVIPS and BSADP under two energy consumption formulations, i.e. ECF-I and ECF-II. The experimental results of Section 4.3 and Section 4.4 can be summarized in Fig. 8. Fig. 8 shows the obtained overall ECD according to the data from Figs. 4-7. Note that, the overall ECD in Fig.8 is a superposition of the basic data. Take Fig. 8 (a) as an example, the curve in Fig. 8 (a) is the superposition of five curves in Fig. 4. By observing Fig. 8, the following conclusions can be made reached:

(1) Fig. 8 (a) and Fig. 8 (b) are obtained under ECF-I. As shown in Eq. (13), ECF-I doesn't consider the energy consumed by the UAV during flight. From Fig. 8 (a) and Fig. 8 (b), in terms of ECD, although BSADP outperforms DEVIPS in each test case, this trend is not obvious, which can be explained as follows. According to Eq. (13), the energy consumption of the UAV is related to hovering power and hovering time. Hovering power is the same for BSADP and DEVIPS. Hovering time is determined by the data transmission time of IoT devices. However, without considering the energy consumption of the UAV during flight, it is very difficult to confirm the relationship between the number of stop points and the data transmission time of IoT devices. In other words, the superiority of BSADP in designing the flight path planning of the UAV can't be displayed fully under ECF-I.

(2) Fig. 8 (c) and Fig. 8 (d) are achieved under ECF-II. Unlike ECF-I, ECF-II considers the energy consumed by the UAV during the flight as shown in Eq. (14). As can be seen from

Fig. 8 (c) and Fig. 8 (d), BSADP can beat DEVIPS in each test case. In addition, with the increase in the number of IoT devices, the advantage of BSADP over DEVIPS becomes gradually larger, which is a very obvious trend, which can be explained as follows. With the increase in the number of IoT devices, more stop points are needed for the UAV, which will lead to an increase in the energy consumption of the UAV during the flight under ECF-II. Benefiting from the excellent ability of BSADP in designing the flight path planning of the UAV, BSADP can achieve a better ECD with the increase in the number of IoT devices compared with DEVIPS.

Based on the above discussion, the improved strategies in BSADP are very effective, which can increase the chance of BSADP escaping from the local optimal solution and help to design the flight path of the UAV.

5 Conclusions and further work

To optimize the energy consumption of a UAV-based IoT data collection system, this paper reported a new variant of backtracking search algorithm called the backtracking search algorithm with a dynamic population (BSADP). Note that, BSADP only needs the essential population size and terminal condition for optimization. To find the optimal energy consumption of the considered system, the population in BSADP is regarded as an entire deployment of the UAV. BSADP has two core components: enhanced backtracking search algorithm (EBSA) and population adjustment mechanism with opposition-based learning (PAMOBL). EBSA is employed to obtain the trail deployment of the UAV and PAMOBL is used for generating the next generation deployment of the UAV by adjusting the trail deployment and the current deployment of the UAV. A series of experiments have been performed to verify the performance of BSADP. Experimental results show the superiority of BSADP in optimizing the energy consumption of the UAV-based IoT data collection system.

The considered system in this work is idealized, which doesn't refer to complex factors, such as the consumed energy to avoid the obstacles in the air. Note that, these complex factors need to be addressed in a real scenario to evaluate accurately the energy consumption of the whole system. Motivated by this, our further work will mainly focus on the following three aspects. Firstly, we will try to build more complex UAV-based IoT data collection systems, which consider more practical factors impacting the energy consumption of this system. Then, how to avoid the obstacles from the air to ensure the fight safety is another focus. Lastly, we will do some attempts to employ reinforcement learning and transfer learning to further improve the global search ability of BSADP.

Compliances with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

B. Hammi, A. Fayad, R. Khatoun, S. Zeadally, Y. Begriche, 2020. A Lightweight ECC-Based Authentication Scheme for Internet of Things (IoT). IEEE Syst. J. 14, 3440–3450. https://doi.org/10.1109/JSYST.2020.2970167

Belhadi, A., Djenouri, Y., Nørvåg, K., Ramampiaro, H., Masseglia, F., Lin, J.C.-W., 2020. Space–time series clustering: Algorithms, taxonomy, and case study on urban smart cities. Eng. Appl. Artif. Intell. 95, 103857. https://doi.org/10.1016/j.engappai.2020.103857

C. Zhan, Y. Zeng, R. Zhang, 2018. Energy-Efficient Data Collection in UAV Enabled Wireless Sensor Network. IEEE Wirel. Commun. Lett. 7, 328–331. https://doi.org/10.1109/LWC.2017.2776922

Civicioglu, P., 2013. Backtracking Search Optimization Algorithm for numerical optimization problems. Appl. Math. Comput. 219, 8121–8144. https://doi.org/10.1016/j.amc.2013.02.017

Derrac, J., García, S., Molina, D., Herrera, F., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. 1, 3–18. https://doi.org/10.1016/j.swevo.2011.02.002

Duan, Y., Lu, Z., Zhou, Z., Sun, X., Wu, J., 2019. Data Privacy Protection for Edge Computing of Smart City in a DIKW Architecture. Eng. Appl. Artif. Intell. 81, 323–335. https://doi.org/10.1016/j.engappai.2019.03.002

Fan, Q., Huang, H., Yang, K., Zhang, S., Yao, L., Xiong, Q., 2021. A modified equilibrium optimizer using opposition-based learning and novel update rules. Expert Syst. Appl. 170, 114575. https://doi.org/10.1016/j.eswa.2021.114575

H. Huang, A. V. Savkin, 2019. A Method for Optimized Deployment of Unmanned Aerial Vehicles for Maximum Coverage and Minimum Interference in Cellular Networks. IEEE Trans. Ind. Inform. 15, 2638–2647. https://doi.org/10.1109/TII.2018.2875041

Houssein, E.H., Rezk, H., Fathy, A., Mahdy, M.A., Nassef, A.M., 2022. A modified adaptive guided differential evolution algorithm applied to engineering applications. Eng. Appl. Artif. Intell. 113, 104920. https://doi.org/10.1016/j.engappai.2022.104920

J. Zhang, A. C. Sanderson, 2009. JADE: Adaptive Differential Evolution With Optional External Archive. IEEE Trans. Evol. Comput. 13, 945–958. https://doi.org/10.1109/TEVC.2009.2014613

Khan, A.I., Alsolami, F., Alqurashi, F., Abushark, Y.B., Sarker, I.H., 2022. Novel energy management scheme in IoT enabled smart irrigation system using optimized intelligence methods. Eng. Appl. Artif. Intell. 114, 104996. https://doi.org/10.1016/j.engappai.2022.104996

M. Jiang, Y. Li, Q. Zhang, J. Qin, 2019. Joint Position and Time Allocation Optimization of UAV Enabled Time Allocation Optimization Networks. IEEE Trans. Commun. 67, 3806–3816. https://doi.org/10.1109/TCOMM.2019.2896973

M. Mozaffari, W. Saad, M. Bennis, M. Debbah, 2017. Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications. IEEE Trans. Wirel. Commun. 16, 7574–7589. https://doi.org/10.1109/TWC.2017.2751045

Marinho, Y.Q., Fruett, F., Giesbrecht, M., 2021. Application of differential evolution to multi-objective tuning of vibration spectrum analyzers based on microelectromechanical systems. Eng. Appl. Artif. Intell. 97, 104071. https://doi.org/10.1016/j.engappai.2020.104071

N. Kaur, S. K. Sood, 2017. An Energy-Efficient Architecture for the Internet of Things (IoT). IEEE Syst. J. 11, 796–805. https://doi.org/10.1109/JSYST.2015.2469676

P. Huang, Y. Wang, K. Wang, K. Yang, 2019. Differential Evolution With a Variable Population Size for Deployment Optimization in a UAV-Assisted IoT Data Collection System. IEEE Trans. Emerg. Top. Comput. Intell. 1–12. https://doi.org/10.1109/TETCI.2019.2939373

Qureshi, K.N., Iftikhar, A., Bhatti, S.N., Piccialli, F., Giampaolo, F., Jeon, G., 2020. Trust management and evaluation for edge intelligence in the Internet of Things. Eng. Appl. Artif. Intell. 94, 103756. https://doi.org/10.1016/j.engappai.2020.103756

Rahnamayan, S., Tizhoosh, H.R., Salama, M.M.A., 2008. Opposition-Based Differential Evolution. IEEE Trans. Evol. Comput. 12, 64–79. https://doi.org/10.1109/TEVC.2007.894200

S. Chou, A. Pang, Y. Yu, 2020. Energy-Aware 3D Unmanned Aerial Vehicle Deployment for Network Throughput Optimization. IEEE Trans. Wirel. Commun. 19, 563–578. https://doi.org/10.1109/TWC.2019.2946822

Sarkhel, R., Das, N., Saha, A.K., Nasipuri, M., 2018. An improved Harmony Search Algorithm embedded with a novel piecewise opposition based learning algorithm. Eng. Appl. Artif. Intell. 67, 317–330. https://doi.org/10.1016/j.engappai.2017.09.020

Tao, F., Jiang, L., Li, C., 2021. Differential evolution-based weighted soft majority voting for crowdsourcing. Eng. Appl. Artif. Intell. 106, 104474. https://doi.org/10.1016/j.engappai.2021.104474

Wang, B.-C., Li, H.-X., Li, J.-P., Wang, Y., 2019. Composite Differential Evolution for Constrained Evolutionary Optimization. IEEE Trans. Syst. Man Cybern. Syst. 49, 1482–1495. https://doi.org/10.1109/TSMC.2018.2807785

X. Li, H. Yao, J. Wang, X. Xu, C. Jiang, L. Hanzo, 2019. A Near-Optimal UAV-Aided Radio Coverage Strategy for Dense Urban Areas. IEEE Trans. Veh. Technol. 68, 9098–9109. https://doi.org/10.1109/TVT.2019.2927425

Xia, X., Gui, L., Yu, F., Wu, H., Wei, B., Zhang, Y.-L., Zhan, Z.-H., 2019. Triple Archives Particle Swarm Optimization. IEEE Trans. Cybern. 1–14. https://doi.org/10.1109/TCYB.2019.2943928

Y. Wang, H. Liu, H. Long, Z. Zhang, S. Yang, 2018. Differential Evolution With a New Encoding Mechanism for Optimizing Wind Farm Layout. IEEE Trans. Ind. Inform. 14, 1040–1054. https://doi.org/10.1109/TII.2017.2743761

Y. Wang, Z. Ru, K. Wang, P. Huang, 2019. Joint Deployment and Task Scheduling Optimization for Large-Scale Mobile Users in Multi-UAV-Enabled Mobile Edge Computing. IEEE Trans. Cybern. 1–14. https://doi.org/10.1109/TCYB.2019.2935466

Yang, Y., Gao, Y., Tan, S., Zhao, S., Wu, J., Gao, S., Zhang, T., Tian, Y.-C., Wang, Y.-G., 2022. An opposition learning and spiral modelling based arithmetic optimization algorithm for global continuous optimization problems. Eng. Appl. Artif. Intell. 113, 104981. https://doi.org/10.1016/j.engappai.2022.104981

Zhao, F., Zhao, J., Wang, L., Cao, J., Tang, J., 2021. A hierarchical knowledge guided backtracking search algorithm with self-learning strategy. Eng. Appl. Artif. Intell. 102, 104268. https://doi.org/10.1016/j.engappai.2021.104268